IRE
Transactions
on ELECTRONIC COMPUTERS

Volume EC-10 **September, 1961** **Number 3**

Published Quarterly

TABLE OF CONTENTS

PUBLISHED BY THE
ofessional Group on ELECTRONIC COMPUTERS

IRE Transactions® on Electronic Computers

Published by The Institute of Radio Engineers, Inc., for the Professional Group on Electronic Computers at 1 East 79 Street, New York 21, N.Y. Responsibility for the contents rests upon the authors and not upon the IRE, the Group, or its members. Individual copies of this issue and all available back issues may be purchased at the following prices: IRE members (one copy) $2.25, libraries and colleges $3.25, all others $4.50. Annual subscription price: libraries and colleges $12.75; non-members $17.00. Address requests to the Institute of Radio Engineers, 1 East 79 Street, N.Y. 21, N.Y.

Notice to Authors: Address all papers and editorial correspondence to the appropriate Editor. Addresses are listed on page 343. To avoid delay and inconvenience in the processing of manuscripts, please follow the procedure suggested there.

Editorial

WITH this issue, the IRE TRANSACTIONS ON ELECTRONIC COMPUTERS appears under new editorship. The first privilege of your new Editor is to record here the appreciation of the members of the Professional Group on Electronic Computers to the retiring Editor, Dr. Howard E. Tompkins, who has served in this post for the past two and one-half years. The high standards of this journal, its professional stature, its broad coverage, and its rapid growth are due in large part to the conscientious and enthusiastic guidance given by Dr. Tompkins.

Under Dr. Tompkins' editorship, the total annual number of pages in the TRANSACTIONS ON ELECTRONIC COMPUTERS nearly doubled. This is evidence not only of the growth of the computer engineering profession but also of the growth of the PGEC as a voice of that profession. To handle the increased pace of publishing activity, Dr. Tompkins made several innovations which are now in the process of being implemented. One of these steps is the plan which was announced in the June issue to change the TRANSACTIONS from a quarterly publication to a bimonthly one. In 1962 the TRANSACTIONS ON ELECTRONIC COMPUTERS will appear in February and every second following month.

A second step is the appointment of a staff of Associate Editors, each specializing in one broad area, to help with the editorial function. The first of these, John E. Sherman, is already serving as Associate Editor for Analog and Hybrid Computers. It is a pleasure to announce at this time the appointment of another Associate Editor, Dr. Edward J. McCluskey, Associate Professor of Electrical Engineering at Princeton University, who will serve as Associate Editor for Logic and Switching Theory, starting at once. In taking up his new post, Dr. McCluskey vacates the position of Reviews Editor, which he has so ably filled, and turns it over to our new Reviews Editor, Thomas C. Bartee of Lincoln Laboratories. Photos and biographies of Dr. McCluskey and Mr. Bartee appear in the News and Notices section of this issue.

Your editorial staff will earnestly try to maintain the high standards of the IRE TRANSACTIONS ON ELECTRONIC COMPUTERS. We extend our thanks in advance to all our readers who help in this effort by submitting papers and by serving as referees.

NORMAN R. SCOTT

An Algorithm for Path Connections and Its Applications*

C. Y. LEE†, MEMBER, IRE

Summary—The algorithm described in this paper is the outcome of an endeavor to answer the following question: Is it possible to find procedures which would enable a computer to solve efficiently path-connection problems inherent in logical drawing, wiring diagramming, and optimal route finding? The results are highly encouraging. Within our framework, we are able to solve the following types of problems:

1) To find a path between two points so that it crosses the least number of existing paths.
2) To find a path between two points so that it avoids as much as possible preset obstacles such as edges.
3) To find a path between two points so that the path is optimal with respect to several properties; for example, a path which is not only one of those which cross the fewest number of existing paths, but, among these, is also one of the shortest.

The minimal-distance solution has been programmed on an IBM 704 computer, and a number of illustrations are presented. The class of problems solvable by our algorithm is given in a theorem in Section III. A byproduct of this algorithm is a somewhat remote, but unexpected, relation to physical optics. This is discussed in Section VI.

I. Introduction

IN processing information consisting of patterns, rather than numbers or symbols, on a digital computer, we may wish to know how a computer, without sight and hearing, can be made to deal competently with situations which appear to require coordination, insight, and perhaps intuition. It is not our intention to consider the general problem of pattern detection and recognition by machines. We will consider rather the following simpler, and therefore perhaps more basic, problem in pattern processing by machines.

Let a pattern of some sort be presented to a machine. We then want the machine to construct some optimal path subject to various constraints imposed by the pattern. The problem is to find efficient procedures, which, if followed by the machine, would lead to an optimal solution.

Ideally, many situations would fall within this description. We might present to the machine a map of Manhattan and ask it to find the shortest-time route between, say, the United Nations and Yankee Stadium, using only public transportation. With sufficient care, it is possible to make a problem such as this unambiguous. In most cases, however, it would be too great a struggle just to present the problem in a way that is completely and consistently stated. For this reason, we have decided to present an abstract model in Section II. Based on this model, we will consider a class of well-defined optimal path problems. A general procedure for solving this class of problems will then be given in Section III.

Within this class of problems is the *shortest-route problem* on which there has been earlier definitive work. Algorithms for finding shortest paths have been given by Dantzig,[1] Ford and Fulkerson,[2] Moore[3] and Prim.[4] The minimal-distance illustrations (Section V) make use of one of Moore's algorithms, which is a specialization of algorithm A given in Section III. These experiments were tried out before the abstract model was completed. Once we have at our disposal the abstract model, it came as a pleasant surprise that problems such as the minimal-crossing and minimal edge-effect problems, which had appeared difficult to us previously, all yielded immediately to algorithm A. The possibility of joint minimization is also a direct consequence of algorithm A. A further outcome was the "diffraction" patterns. These experiments would not have been attempted if we had not noticed the patterns obtained in the minimal-distance experiments.

II. An Abstract Model and the Path Problem

A. $\mathcal{C}$-Space, an Abstract Model

Let C be a set of elements called cells: $C = \{c^1, c^2, \cdots\}$. For each cell c^i in C, there is defined a subset of C called a *1-neighborhood* $N(c^i)$ of c^i: $N(c^i) = \{c_1^i, c_2^i, \cdots, c_n^i\}$. The following rules hold for 1 neighborhoods:

N1) Every 1-neighborhood has in it exactly n cells, where n, $n \geq 1$, is some predetermined number depending on the specific model involved.

N2) If $c^i \in N(c^i)$, then $c^i \in N(c^i)$. We will call the function N with domain C and range subsets of C the *1-neighborhood function.*

Together with the 1-neighborhood functions, there are n functions $d_1, d_2, \cdots, d_n$ on C to C defined as follows: If c^i is any cell in C and $N(c^i) = \{c_1^i, c_2^i, \cdots, c_n^i\}$, then

$$d_k(c^i) = c_k^i, \qquad k = 1, 2, \cdots, n.$$

* Received by the PGEC, December 2, 1960. This material was presented as part of the University of Michigan Engineering Summer Session, Ann Arbor, June 19–30, 1961.

† Bell Telephone Labs., Inc., Whippany, N. J.

[1] G. B. Dantzig, "Maximization of a Linear Function of Variables Subject to Linear Inequalities," Cowles Commission; 1951.

[2] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Can. J. Math.*, vol. 8, pp. 399–404; 1956.

[3] E. F. Moore, "Shortest path through a maze," in "Annals of the Computation Laboratory of Harvard University," Harvard University Press, Cambridge, Mass., vol. 30, pp. 285–292; 1959.

[4] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Sys. Tech. J.*, vol. 36, pp. 1389–1401; November, 1957.

That is, $d_k(c^i)$ is the kth coordinate cell in the 1-neighborhood of c^i.

Let S be a finite set of symbols: $S=\{s^1, s^2, \cdots, s^m\}$. S is called the *alphabet* of space $\mathcal{C}$.

Let Γ be a map on C to $C\times S$. That is, for every $c^i\in C$, $\Gamma(c^i)=(c^i, s^i)$, $s^i\in S$. In general we will write $\Gamma(c^i)=(c^i, s(c^i))$. Therefore, to every cell $c^i\in C$ is associated some symbol $s(c^i)\in S$. The map Γ gives then the cell-symbol configuration for a particular path problem. Generally speaking, we may keep in mind the analogy that each function Γ corresponds to some sort of a street map for a particular city or town, and that the path problem is to find an optimal path from one point to another in that city or town.

Let c^i, c^j be two distinct cells in C. By a *path* $p(c^i, c^j)$ is meant a set of cells called a *chain:* $p(c^i, c^j)=\{c^0=c^i, c^1, c^2, \cdots, c^m=c^j\}$ such that $c^{i+1}\in N(c^i)$ for $i=0, 1, \cdots, m-1$. By $\pi(c^i, c^j)$ is meant the set of all paths $p(c^i, c^j)$ between cell c^i and cell c^j.

Let M be a map called an *admission map* with domain $\pi(c^i, c^j)$ and range the two-element set $\{0, 1\}$. Any path $p(c^i, c^j)$ such that $M(p(c^i, c^j))=1$ is said to be an *admissible path.* Otherwise, $p(c^i, c^j)$ is said to be *inadmissible.* The set of all admissible paths will be denoted by $\pi^*(c^i, c^j)$.

The quintuple (C, S, N, Γ, M) is called a $\mathcal{C}$-*space.*

B. The Path Problem

Let F be a vector of r functions $(f_1, f_2, \cdots, f_r)$ where each function f_i, $i=1, 2, \cdots, r$, is on $\pi^*(c^i, c^j)$, the set of admissible paths, to I, the set of non-negative integers. A path $p^1(c^i, c^j)$ of $\pi^*(c^i, c^j)$ is said to be *minimal with respect to* f_1 if

$$f_1(p^1(c^i, c^j)) \leq f_1(p(c^i, c^j))$$

for all $p(c^i, c^j)\in\pi^*(c^i, c^j)$. A path $p^{12}(c^i, c^j)$ is said to be *minimal with respect to* (f_1, f_2) if

$$\text{(i)}\quad p^{12}(c^i, c^j) \in P^1(c^i, c^j)$$

where $P^1(c^i, c^j)$ is the set of all paths in $\pi^*(c^i, c^j)$, which are minimal with respect to f_1; that is,

$$P^1(c^i, c^j) = \{p^1(c^i, c^j) \mid f_1(p^1(c^i, c^j)) \leq f_1(p(c^i, c^j)) \text{ for all } p(c^i, c^j) \in \pi^*(c^i, c^j)\};$$

and

$$\text{(ii)}\quad f_2(p^{12}(c^i, c^j)) \leq f_2(p(c^i, c^j)) \text{ for all } p(c^i, c^j) \in P^1(c^i, c^j).$$

Therefore, $p^{12}(c^i, c^j)$ is minimal with respect to (f_1, f_2) if among all paths minimal with respect to f_1, $p^{12}(c^i, c^j)$ is also minimal with respect to f_2. In a similar way, we may define a path $p^{12\cdots r}(c^i, c^j)$ which is minimal with respect to $(f_1, f_2, \cdots, f_r)$.

The path problem we are considering is the following:

P) Path problem: Given a $\mathcal{C}$-space (C, S, N, Γ, M), a vector $F=\{f_1, f_2, \cdots, f_r\}$, an initial cell c^* and a final cell c^{**}, find an admissible path $p^{12\cdots r}(c^*, c^{**})$ which is minimal with respect to $(f_1, f_2, \cdots, f_r)$.

In the following section we will show an algorithm which solves the path problem P for a certain set of vectors F.

III. A Search and Trace Procedure

A. Montone Functions and Monotone Vectors

Let a $\mathcal{C}$-space (C, S, N, Γ, M) be given. A function f on $\pi^*(c^i, c^j)$ to the set of non-negative integers I is said to be *monotone* if for every path $p(c^i, c^j)$, we have the inequality

$$f(p(c^i, c^k)) \leq f(p(c^i, c^j)),$$

where $p(c^i, c^k)$ is any subpath of $p(c^i, c^j)$. A vector F of monotone functions $(f_1, f_2, \cdots, f_r)$ is said to be a *monotone vector.*

B. An Algorithm

Let c^* and c^{**} be the initial and final cells in a $\mathcal{C}$-space (C, S, N, Γ, M). The main procedure will be given in terms of a number of subprocedures.

D1) *Cell list:* A *cell list* L is an ordered list containing names of cells.

D2) *Cell mass:* With each cell in list L will be associated an r-tuple called a *cell mass* $(m_1, m_2, \cdots, m_r)$. The cell masses are ordered lexicographically. Thus, if $m=(m_1, m_2, \cdots, m_r)$, and $m'=(m_1', m_2', \cdots, m_r')$ are two cell masses, $(m_1, m_2, \cdots, m_r)<(m_1', m_2', \cdots, m_r')$ if $m_i=m_i'$, $i=0, 1, \cdots, k$, but $m_{k+1}<m'_{k+1}$; $0\leq k<r$.

D3) *Chain coordinate:* Associated with each cell in the list L is also a *chain coordinate* which is one of the 1-neighborhood coordinate functions d_k.

D4) *Auxiliary list* L_1: An auxiliary cell list L_1 is provided for momentary storage of names of cells.

R1) *Procedure for constructing auxiliary list* L_1: Let c be a cell in list L. By a path $p(c^*, c^i, c^{**})$ we mean a path $p(c^*, c^{**})$ of which $p(c^*, c^i)$ is a subpath. A cell c^i is said to be *admissible* if $p(c^*, c^i, c^{**})$ is an admissible path and if the cell mass for c^i has not yet been determined. Let $\{c^i\}\in N(c)$ be the set of all admissible cells in $N(c)$. Append to list L_1 the set $\{c^i\}$. L_1 is constructed by repeating this process for every entry c in L, under the condition that a cell should not be listed more than once. L_1 is therefore the set of all distinct cells c^i such that c^i is an admissible cell in $N(c)$ for some cell c in the list L.

R2) *Procedure for assigning cell masses and chain coordinates:* Let c^i be a cell in L_1. A *possible cell mass* for c^i is determined as follows: For each $c^j\in N(c^i)$ whose cell mass has been determined, construct an r-tuple,

$$(f_1(p(c^*, c^j, c^i)), \cdots, f_r(p(c^*, c^j, c^i))).$$

Now apply rule R3 below to find a $c^{j_0}\in N(c^i)$, for which this r-tuple is a minimum. A possible cell mass for c^i is then the r-tuple $(f_1(p(c^*, c^{j_0}, c^i)), \cdots, f_r(p(c^*, c^{j_0}, c^i)))$, and a possible chain coordinate for c^i is d_k where $d_k(c^i)=c^{j_0}$.

Next, find possible cell masses for all $c^i \in L_1$. Among the cells in L_1, let $\{c^l\}$ be the set of cells whose possible cell masses are a minimum. We then assign to each $c^l \in \{c^l\}$ the same cell mass and the same chain coordinate as its possible cell mass and its possible chain coordinate.

The cells whose possible cell masses are not minimal are no longer considered, and the list L_1 is cleared.

R3) *Selection rule:* In R2, a cell c^i and a selected subset $\{c^j\}$ of $N(c^i)$ were given. The rule which was invoked to select one c^{j_0} from the set $\{c^j\}$ is called a *selection rule.*

R4) *Procedure for updating list L:* Let $\{c^l\}$ be the set of cells whose cell masses had been determined by R2. Append first to list L the set $\{c^l\}$. Next examine each cell c in L to see if the cell masses of all its admissible 1 neighbors had been determined. If so, erase c from the list L.

R5) *Initialization:* Cell c^* is given the cell mass $(0, 0, \cdots, 0)$. The list L contains initially the single entry c^*. The list L_1 is initially cleared.

With the aid of rules R1–R5, we may now state the main procedure.

A: The Search and Trace Algorithms:

A1: Search algorithm: Apply rule R5 for initialization. Apply rules R1, R2, R4 to entries of L until either c^{**} appears in L or the list L has been exhausted. In the former case, we will proceed to the trace algorithm A2. In the latter case, there is no admissible path $p(c^*, c^{**})$ in the $\mathfrak{C}$ space.

A2: Trace algorithm: Begin at c^{**}, follow the chain coordinates until c^* is reached. This determines a unique path $p(c^*, c^{**})$.

C. The Procedure Applied to the Path Problem

Let a $\mathfrak{C}$ space (C, S, N, Γ, M), a vector $F = (f_1, f_2, \cdots, f_r)$ and an initial cell c^* be given. A cell c^i which is reached from c^* after exactly t application of rules R1 to R4 is said to have a *chain index* l (c^*, c^i) of t from c^*. We will begin with two lemmas.

Lemma 1: Let F be a monotone vector. Let c^i and c^j be two cells with cell masses $m(c^i)$ and $m(c^j)$ and chair indexes $\ell(c^*, c^i)$ and $\ell(c^*, c^j)$, respectively. If $m(c^i) < m(c^j)$, then $\ell(c^*, c^i) < \ell(c^*, c^j)$.

Proof: Assume $\ell(c^*, c^i) \geqq \ell(c^*, c^j)$. This means that after $\ell(c^*, c^j)$ applications of rules $R1$ to $R4$, the cell mass $m(c^j)$ has been determined, but the cell mass $m(c^i)$ has not. By the nature of rule $R2$, the cell masses are obtained by evaluating the vector F for the paths in question. Since F is monotone by hypothesis, $m(c^i) \geqq m(c^j)$ and the lemma follows.

Lemma 2: Let F be a monotone vector. Let $p(c^*, c^{**})$ be any admissible path from c^* to c^{**}. Then $m(c^{**}) \leqq F(c^*, c^{**})$.

Proof: Let $p(c^*, c^{**})$ consist of the chain of cells

$$p(c^*, c^{**}) = \{c^* = c^0, c^1, c^2, \cdots, c^q, c^{**} = c^{q+1}\}.$$

We will let the number of cells contained in a path, excluding the initial cell, be called the *path length* of that path. For the path $p(c^*, c^{**})$, therefore, the path length is $q+1$.

For admissible paths of length 1, it follows from rule $R2$ that $m(c^{**}) \leqq F(c^*, c^{**})$. Let us now assume valid the induction hypothesis that $m(c^{**}) \leqq F(c^*, c^{**})$ for all admissible paths of path length not greater than q.

Consider the subpath $p(c^*, c^q)$ of $p(c^*, c^{**})$. By the induction hypothesis,

$$m(c^q) \leqq F(c^*, c^q).$$

Let us now suppose $m(c^{**}) > F(c^*, c^{**})$. Since F is monotone, we then have the inequalities

$$m(c^q) \leqq F(c^*, c^q) \leqq F(c^*, c^{**}) < m(c^{**}).$$

Since now $m(c^q) < m(c^{**})$, it follows from Lemma 1 that $\ell(c^*, c^q) < \ell(c^*, c^{**})$. Therefore the cell mass $m(c^q)$ is determined before the cell mass $m(c^{**})$ is determined.

From rule $R2$ it follows that as soon as $m(c^q)$ is determined, the cell c^q becomes a member of cell list L. Since by rule $R2$,

$$m(c^{**}) = \begin{cases} \min \{F(p(c^*, c^i, c^{**})) \mid c^i \in N(c^{**})\} \\ \text{over all } c^i \text{ for which } m(c^i) \text{ is defined,} \end{cases}$$

and since c^q is one of such cells c_i, it follows that this minimum cannot be greater than $F(p(c^*, c^{**}))$. This contradicts our earlier supposition, and the lemma follows.

The basic result is embodied in the following:

Theorem: Let a $\mathfrak{C}$-space (C, S, N, Γ, M) be given. Let P be a path problem with respect to a vector F. If F is monotone, then algorithm A yields a path $\bar{p}(c^*, c^{**})$ satisfying P.

Proof: Let $p(c^*, c^{**})$ be any admissible path from c^* to c^{**}. Since F is monotone, we may apply Lemma 2 to get $m(c^{**}) \leqq F(p(c^*, c^{**}))$. It follows from rule R2 that

$$m(c^{**}) = (f_1(\bar{p}(c^*, c^{**})), \cdots, f_r(\bar{p}(c^*, c^{**}))).$$

Also, by definition,

$$F(p(c^*, c^{**})) = (f_1(p(c^*, c^{**})), \cdots, f_r(p(c^*, c^{**}))).$$

The theorem therefore follows by the lexicographic ordering of these r-tuples.

IV. Applications

In the applications to be discussed here, we will consider the set C to be a set of squares in the plane with the usual 1 neighborhoods as shown in Fig. 1. We will let the alphabet set S consist of the following:

1) Digits from 0 to 9.
2) All letters of the English alphabet.
3) The symbols: +, −, ·, ,, /, *, —, └, ┐, ┘, ┌, (,), →, ←, ↑, ↓, blank.

The 1-neighborhood function N and the coordinate functions d_1, d_2, d_3, and d_4 are defined as follows: Given a cell c^i, $d_1(c^i)$ is the cell to the right of c^i, $d_2(c^i)$ is the cell above c^i, $d_3(c^i)$ is the cell to the left of c^i and $d_4(c^i)$

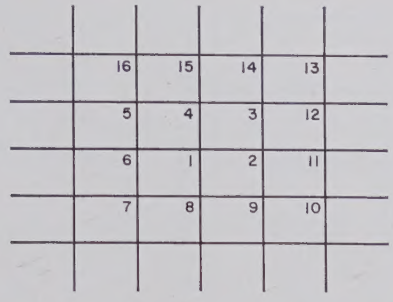


Fig. 1—The set C of squares in the plane.

is the cell below c^i. $N(c^i)$ is then the set $\{d_1(c^i), d_2(c^i), d_3(c^i), d_4(c^i)\}$.

Let a function Γ be given, so that to each cell $c^i \in C$ is associated a symbol $s(c^i) \in S$. The function M is given as follows:

Let $p(c^*, c^{**}) = \{c^0 = c^*, c^1, \cdots, c^{n-1}, c^n = c^{**}\}$ be a path from c^* to c^{**}. $p(c^*, c^{**})$ is admissible, *i.e.*, $M(p(c^*, c^{**})) = 1$, if

1) $s(c^i) =$ blank, —, or $|$ for $i = 1, 2, \cdots, n-1$.
2) $s(c^{i+1}) \neq$ — whenever $c^{i+1} = d_1(c^i)$ or $c^{i+1} = d_3(c^i)$, $i = 0, 1, \cdots, n-1$.
3) $s(c^{i+1}) \neq |$ whenever $c^{i+1} = d_2(c^i)$ or $c^{i+1} = d_4(c^i)$, $i = 0, 1, \cdots, n-1$.

Thus, except for the function Γ which depends on the application in question, the specialization of the $\mathfrak{C}$-space to our applications has been fully described.

A. A Minimal-Crossing Problem

Given a set of squares in the plane, the problem of finding a path $p(c^*, c^{**})$ from c^* to c^{**} such that $p(c^*, c^{**})$ crosses over the fewest number of existing paths is called the minimal-crossing problem. We will formulate the minimal-crossing problem as a path problem in some appropriate $\mathfrak{C}$ space and then solve it with the aid of algorithm A.

For this problem the vector F would consist of a single function f given as follows:

F1) $f(p(c^*, c^*)) = 0$.

F2) If $s(c^i) =$ blank, then

$$f(p(c^*, c^i)) = \begin{cases} \min \{f(p(c^*, c^i)) \mid c^i \in N(c^i)\} \text{ over all } c^i \\ \quad \text{for which } f(p(c^*, c^i)) \text{ has been defined;} \\ \text{undefined otherwise.} \end{cases}$$

F3) If $s(c^i) =$ —, then

$$f(p(c^*, c^i)) = \begin{cases} (\min\{f(p(c^*, d_2(c^i))), f(p(c^*, d_4(c^i)))\}) + 1 \\ \quad \text{if either one of the values of } f \text{ is defined;} \\ \text{undefined otherwise.} \end{cases}$$

F4) If $s(c^i) = |$, then

$$f(p(c^*, c^i)) = \begin{cases} (\min\{f(p(c^*, d_1(c^i))), f(p(c^*, d_3(c^i)))\}) + 1 \\ \quad \text{if either one of the values of } f \text{ is defined;} \\ \text{undefined otherwise.} \end{cases}$$

We assert first that f is monotone. This is so by the iterative nature of the definition of f; the value of f never decreases as a path increases in length. Hence, we may apply the basic theorem to get:

Corollary 1: Algorithm A solves the minimal-crossing problem.

Example 1: Consider the cell configuration given in Fig. 2. The path AA consisting of the chain of cells $\{6, 5, 16, 15, 14, 3, 2, 11, 28\}$ is already present. We wish to find a minimal-crossing path from c^* (cell 18), to c^{**} (cell 13).

Applying algorithm A, we find that list L consists of the single entry $\{18\}$ to begin with. List L_1 therefore has in it entries $\{5, 17, 19\}$. Note that cell 39 is not an admissible 1-neighbor of c^*. The possible cell masses for cells $\{5, 17, 19\}$ are 1, 0 and 0, respectively. Hence, by rule R2, cells 17 and 19 are assigned cell mass 0 and are appended to the list L. Moreover, the chain coordinates of cells 17 and 19 are respectively d_4 and d_2. The cell masses and chain coordinates for these cells are properly denoted in Fig. 2. Thus, in cell 17, we have the pair $(\downarrow, 0)$, meaning that the chain coordinate is d_4 (*i.e.*, downward) and the cell mass is 0.

Applying rules R1 to R4 again, we find that list L_1 has in it now the entries $\{5, 20\}$. This is so since cells 16, 36, 38, 39, 6 and 40 are all not admissible. The possible cell masses for cells 5 and 20 are respectively 1 and 0. Thus, the cell mass for cell 20 is 0, the chain coordinate for cell 20 is d_2, and cell 20 is appended to list L. Moreover, cells 17 and 19 are erased from list L.

Fig. 3 shows the cell mass and chain coordinates for all the cells reached by the application of Algorithm A. The trace algorithm then traces out a solution path $p(c^*, c^{**})$ as shown. The boundary cells have been omitted in Fig. 3.

Example 2: Consider again the cell configuration in Fig. 2. In this case, however, we stipulate that it costs 3 units to cross a —, but 1 unit to cross a $|$. That is, the definition F3 is changed to read:

F3′. If $s(c^i) =$ —, then

$$f(p(c^*, c^i)) = \begin{cases} (\min\{f(p(c^*, D_2(c^i))), f(p(c^*, D_4(c^i)))\}) + 3 \\ \quad \text{if either one of the values of } f \text{ is defined;} \\ \text{undefined otherwise.} \end{cases}$$

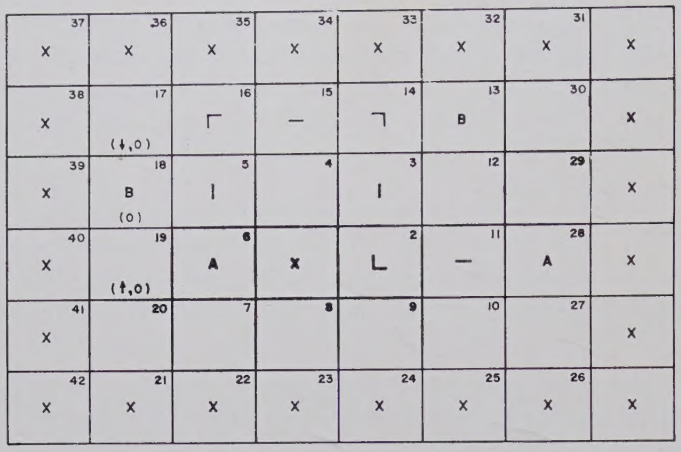


Fig. 2—A minimal-crossing example.

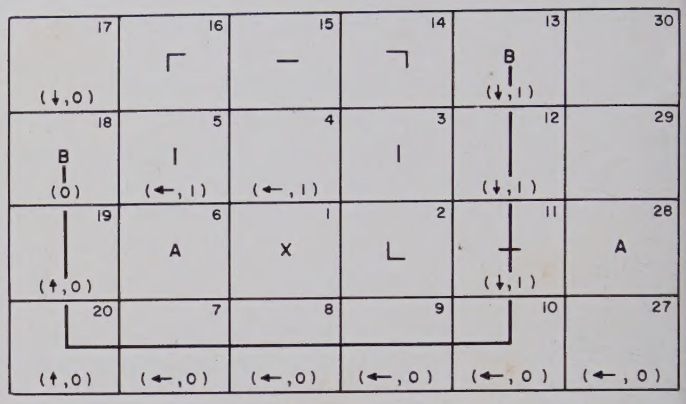


Fig. 3—Cell configuration after algorithm A has been applied to Example 1.

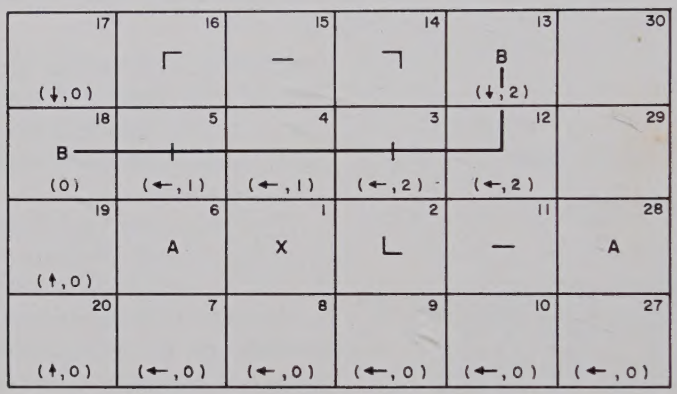


Fig. 4—Cell configuration after algorithm A has been applied to Example 2.

Since F is still monotone, algorithm A may be applied to solve this modified path problem.

In this case, after applying algorithm A, we have the cell configuration and the solution path shown in Fig. 4.

In Example 1, the function f is not only monotone, but grows a single unit at a time. For such functions and for vectors made up of such functions, it is possible to simplify algorithm A to solve the path problem. In Example 2 the function f no longer grows one unit at a time. The machinery of $\mathfrak{C}$-space is needed to cope with this more general class of monotone functions.

B. A Minimal-Edge-Effect Problem

Let us begin with the $\mathfrak{C}$-space consisting of squares in the plane described earlier. Let an initial cell c^* and a final cell c^{**} be given. We wish now to consider the problem of finding a path from c^* to c^{**} which avoids, as much as possible, any symbol other than —, | and the blank symbol. In other words, we want a path which does not tend to "cling to edges."

For this application, the vector F would again consist of a single function g given as follows:

G1) $g(p(c^*, c^*)) = 0$.

G2) If $s(c^i)$ = blank, then

$$g(p(c^*, c^i)) = \begin{cases} (\min \{g(p(c^*, c^j)) \mid C^j \in N(c^i)\}) + R(c^i) \\ \text{over all } C^j \text{ for which } g(p(c^*, c^j)) \text{ has been defined;} \\ \text{undefined otherwise,} \end{cases}$$

where $R(c^i)$ = the number of cells c^j in $N(c^i)$ in each of which the symbol is neither blank, nor —, nor |.

G3) If $s(c^i) =$ —, then

$$g(p(c^*, c^i)) = \begin{cases} (\min \{g(p(c^*, d_2(c^i))), g(p(c^*, d_4)c^i)))\}) + R(c^i) \\ \qquad \text{if either value of } g \text{ is defined;} \\ \text{undefined otherwise:} \end{cases}$$

G4) If $s(c^i) =$ |, then

$$g(p(c^*, c^i)) = \begin{cases} (\min \{g(p(c^*, d_1(c^i))), g(p(c^*, d_3(c^i)))\}) \\ \qquad + R(c^i) \text{ if either value of } g \text{ is defined;} \\ \text{undefined otherwise.} \end{cases}$$

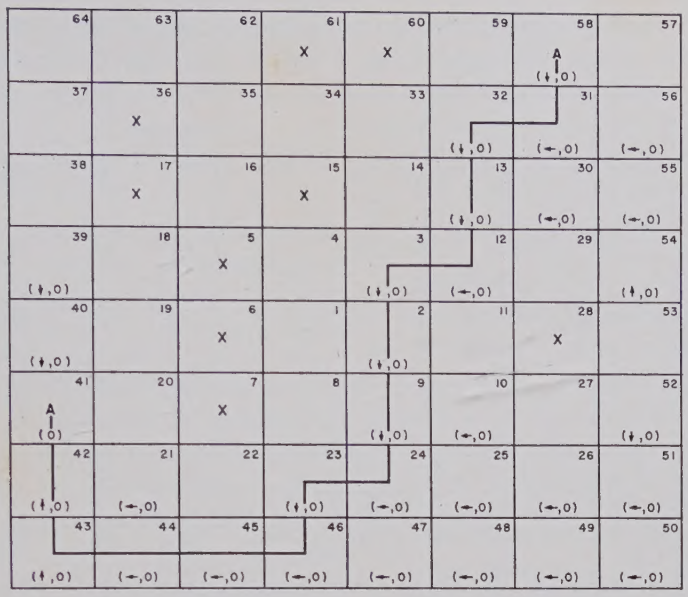


Fig. 5—A minimal edge-effect example.

From this it follows that F is again monotone. Therefore, we have:

Corollary 2: Algorithm A solves the minimal edge-effect problem.

Example 3: Consider the cell configuration given in Fig. 5. We wish to construct a path AA from cell 41 to cell 58 satisfying the path problem with respect to the vector F given by G1 to G4.

To begin with, the list L has in it the single entry $\{41\}$. The list L_1, therefore, has in it entries $\{20, 40, 42\}$. The possible cell masses for these cells are, respectively, 1, 0, 0. Therefore, by rule R2, $m(40) = m(42) = 0$. The chain coordinates for cells 40 and 42 are also determined and are respectively $\downarrow$ and $\uparrow$.

List L is now updated to contain cells $\{41, 40, 42\}$. From this, we get for list L_1 the cells $\{20, 19, 39, 21, 43\}$. Following rule R2, we get therefore $m(39) = m(43) = m(21) = 0$, and also their chain coordinates. By rule R4, cell 42 is erased from list L.

The new list L now has in it cells $\{41, 40, 39, 43, 21\}$. Therefore, cells $\{20, 19, 18, 38, 22, 44\}$ all belong to list L_1. Consider now cell 44. It is clear that $m(44) = 0$. We must, however, invoke the selection rule R3 to get its chain coordinate. We may assume, in this case, that the left neighbor is always preferred over the other 1-neighbors. The chain coordinate for cell 44 then becomes $\leftarrow$.

Continuing in this way, we arrive at the path shown in Fig. 5. We see that the path so constructed avoided all the $\times$'s which may be considered as edges.

C. Joint Minimization

In the last two applications, the vector F in each case consisted of a single function. This was so because we were looking for paths which were minimal with respect to a single property. In the first case, the property was the number of crossings, and in the second, the property was edge effect.

On the other hand, the C-space model was intended to solve joint minimization problems; *i.e.*, the vector F may have several component functions. In order to illustrate the possibility of joint minimization, let us consider a *minimal distance then edge-effect problem.* What we wish to find here is a path which is, first of all, one of the shortest, and secondly, among all shortest paths, this path is also minimal with respect to edge effect. The vector F for this problem has, therefore, two component functions $F = (h, g)$, where h is the distance function and g is the edge-effect function.

The edge-effect function g will be taken to be exactly the same as that defined previously in G1 to G4. The distance function h is given as follows:

H1) The function h satisfies F1, F3, F4.

H2) If $s(c^i) =$ blank, then

$$h(p(c^*, c^i)) = \begin{cases} (\min \{h(p(c^*, c^j)) \mid c^j \in N(c^i)\}) + 1 \text{ over all} \\ \quad c^j \text{ for which } h(p(c^*, c^j) \text{ has been defined;} \\ \text{undefined otherwise.} \end{cases}$$

From these definitions it again follows that both h and g are monotone functions. $F = (h, g)$ is therefore a monotone vector. Therefore, we have:

Corollary 3: Algorithm A solves the joint minimization problem with respect to first distance then edge effect.

Example 4: Let us consider again the original configuration shown in Fig. 5. We wish to find a path AA which solves the path problem with respect to the vector $F = (h, g)$ for this configuration.

Following rule R5, we begin with cell 41 in list L. By rule R1, cells 20, 40 and 42 therefore belong to list L_1. The possible cell masses for cells 20, 40 and 42 are

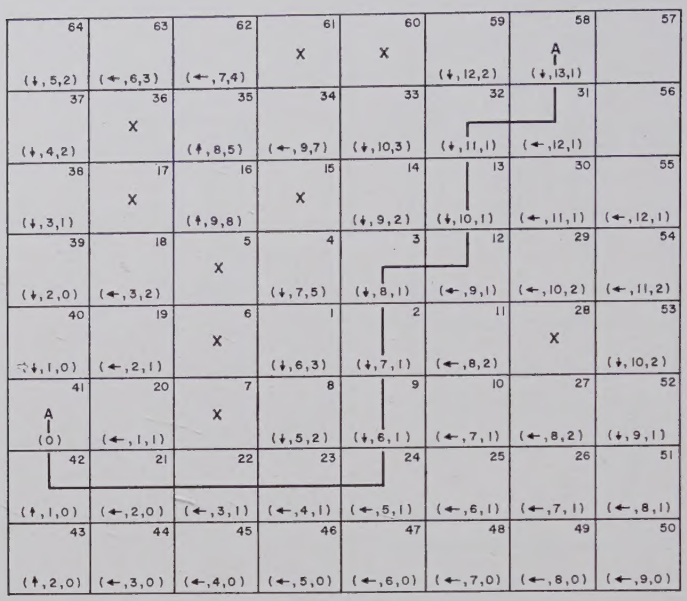


Fig. 6—A joint minimization example.

respectively (1, 1), (1, 0) and (1, 0). Therefore, $m(40) = (1, 0)$ and $m(42) = (1, 0)$. The chain coordinates for cells 40 and 42 are respectively $\downarrow$ and $\uparrow$.

From rule R4, the list L now contains cells 41, 40 and 42. Therefore, by R1, list L_1 has now in it cells $\{20, 19, 39, 43, 21\}$ with possible cell masses respectively: (1, 1), (2, 1), (2, 0), (2, 0), (2, 0). Thus $m(20) = (1, 1)$ and cell 20 has chain coordinate $\leftarrow$.

Continuing in this manner, and using again the selection rule R3 that the left direction is to be preferred over all other directions, we arrive at the path shown in Fig. 6. We see that in this case, since we are interested in the shortest path, the path AA makes contact once with an edge. We had seen before that there are paths which make no contact with any edge.

D. Generalizations

We should, perhaps, take a moment at this time to reflect on the following two questions. In Section II, we have set up an abstract model, our ℭ-space, in such a way that algorithm A can be used to solve path problems formulated within this model. The first question we will ask is whether algorithm A can be applied to still more general situations—that is, whether our abstract model of ℭ-space may be further generalized. In a similar way, we may wish to relax the monotone condition on vectors F. The second question is, therefore, whether our basic theorem may be generalized to include also perhaps a subclass of non-monotone vectors.

The answer to the first question can be given in the affirmative, and may appear a bit surprising. Specifically, we may redefine the 1-neighborhood function N such that N needs to satisfy neither rule $N1$ nor rule $N2$ of Section II. The only condition that N must satisfy is a *finiteness* condition:

NO. Every 1-neighborhood is finite.

Let us call a space (C, S, N^*, Γ, M) a ℭ-*space if the 1-neighborhood funciton N^* satisfies the finiteness condition NO rather than conditions $N1$ and $N2$ of Section II. Accordingly, we must also make minor changes to algorithm A. For instance, we must redefine the coordinate functions d_i, and modify our process of assigning chain coordinates. Let us agree to call the modified algorithm A^*. Then our basic theorem would read:

Let a ℭ*-space (C, S, N^*, Γ, M) be given. Let P be a path problem with respect to a vector F. If F is monotone, then algorithm A^* yields a path $\bar{p}(c^*, c^{**})$ satisfying P.

In regard to the second question, our knowledge is very meager. It is quite possible that an essentially different algorithm is needed to deal with non-monotone vectors.

Coming back to our basic theorem, we ought to make it clear that even when a vector F is monotone, it may be so pathological that the process of applying algorithm A could become extremely tedious. To be specific, let $p(c^1, c^n)$ be an admissible path consisting of the chain of cells:

$$p(c^1, c^n) = \{c^1, c^2, \cdots, c^n\}.$$

A monotone function f is said to be *1-hereditary* if $f(p(c^1, c^n))$ depends only on $f(p(c^1, c^{n-1}))$ and on cells c^{n-1} and c^n. A monotone function f is said to be *2-hereditary* if $f(p(c^1, c^n))$ depends only on $f(p(c^1, c^{n-2}))$ and $f(p(c^1, c^{n-1}))$ and cells c^{n-2}, c^{n-1} and c^n. In a similar manner, we may define p-hereditary functions for $p \geqq 1$.

All of the examples of monotone vectors given in this section happen to be 1-hereditary. In such cases the process of applying algorithm A is much simplified. We may also apply algorithm A to solve the *minimal corner problem*; that is, to find a path with the least number of corners. This problem presents an interesting twist, since the corner function is monotone but 2-hereditary. In the same way, one may construct p-

hereditary functions for arbitrary p. Indeed, one may construct monotone functions which are not finitely hereditary.

V. Minimal-Distance Solutions—A Maze and Other Illustrations

As the reader can see, the statement of algorithm A lends itself quite directly to computer programming. Such a step for the minimal distance problem has been carried out. A natural cell configuration is determined by the printer associated with the computer. The printer can print 120 characters in one line, and usually prints 60 lines to a page. The cell configuration is therefore a rectangular array of 120 cells by 60 cells. The sets of symbols are, in this case, the set of characters on the printer.

In Fig. 7 is shown an input into the computer. The boundary cells and the obstacle cells are all marked $\times$. In this illustration, we wish to find a minimal-distance path between cell A and cell B.

Fig. 8 shows the result of applying algorithm A1 (the search algorithm) to the cell configuration given in Fig. 7. Since it is not possible to print more than one character in each cell, we have chosen to print out the least significant octal digit of the cell masses. The reader can see that the immediate neighbors of cell A has cell mass 1. The neighbors of these have cell mass 2, etc. This search-expansion process continues until cell B is reached.

Fig. 9 shows the result of applying algorithm A2 (the trace algorithm) to the cell configuration. The selection rule used here is to order the admissible neighbors according to the following list of preference: right, up, left, down. The path shown is the machine's solution to the original path problem.

Fig. 10 depicts a three-stage adder circuit; each box designates one of the circuit stages. We wish to apply algorithm A to establish all appropriate connections.

Fig. 11 shows the result of applying algorithm A. One may note that in these paths, there are many more corners than necessary. The appearance of these extra corners is due to our simple selection rule. To a large extent, the appearance of paths can be controlled by incorporating appropriate selection rules into algorithm A.

Shortly after the program was written, we realized that this program, without change, can be used to also solve maze problems. In this sense then, and with proper modifications, the program may serve in particular as a 704 version of Shannon's maze-solving machine.[5] An illustration is the Hampton court maze shown in Fig. 12. Fig. 13 shows the result of applying the search algorithm to the maze configuration. The machine's solution is given by Fig. 14.

VI. The Search Algorithm and Huygens' Principle

In the course of programming algorithm A for the minimal-distance problem, it occurred to us that the search algorithm is, in a remote sense, a computer model of waves expanding from a source under a form of straight-line geometry. Those cells having the same cell mass may be thought of as the locations of the wavefront at the mth unit of time. Fig. 8, for example, may be taken to represent the wavefront originating from source A as it expanded.

Following this line of thought, we proceeded to run a few simple experiments suggested by optics. In Fig. 15, the obstacle consisted of a vertical barrier with a slit in the middle. The source is at the left-hand end of the diagram. The blank spaces may serve as an indication of the propagation of the wavefront. The reader may see that the wave pattern to the right of the obstacle is the same as that which would have been produced by a source located at the slit.

In the same way, Fig. 16 shows the effect of having a 2-slit obstacle and Figs. 17 and 18 are patterns created by having a number of multiple-slit obstacles.

There are several differences between the model shown here and the usual diffraction patterns in optics. The geometry assumed in this model is, in the first place, not Euclidean. Since distances in this geometry are measured roughly as distances are understood by taxi drivers on the island of Manhattan, this geometry is sometimes called *Manhattan geometry*.

In addition to the difference in geometry, and the fact that we are dealing with a discrete space, we have also not taken into account the phenomenon of interference. The computer instead has a built-in first-come-first-served rule. That is, where there are several sources present, the amplitude of the wave at any point is determined by the source closest to it.

Although there is only a remote resemblance between this model and optics, these experiments seem to suggest the possibility of microsimulation of physical phenomena on a computer and the possibility of looking for effects in this way if the laws of nature were modified. In our case, it would be possible to include also interference. The model would then be a reasonably faithful representation of elementary wave phenomena under discrete Manhattan geometry.

VII. Acknowledgment

In the course of this work, the writer has had the benefit of stimulating conversations with a number of people. Among these are E. F. Moore, T. H. Crowley, D. H. Evans, S. H. Washburn, R. W. Hamming and C. A. Lovell. The "diffraction" experiments were an outgrowth of a conversation with D. H. Evans. The writer also wishes to acknowledge the untiring assistance for Sections V and VI from S. O. Four of IBM.

[5] C. E. Shannon, "Presentation of the maze-solving machine," *Trans of the 8th Cybernetics Conf.*, Josiah Macy Jr. Foundation, New York, N. Y., pp. 173–180; 1952.

(*See Figs. 7–18, on following pp. 354–365.*)

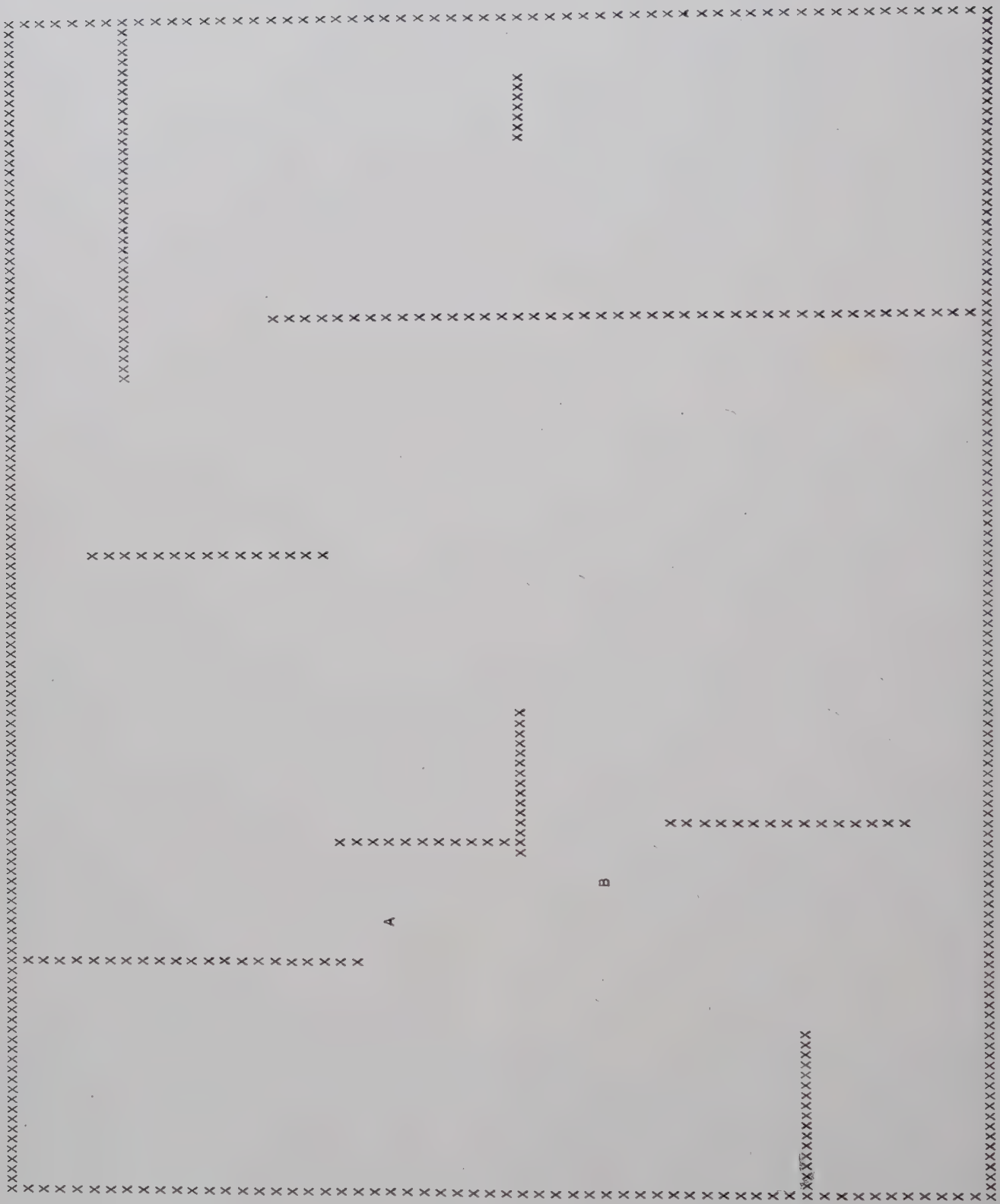


Fig. 7—Input into computer for minimal-distance solutions.

Fig. 8—Result of applying search algorithm to Fig. 7.

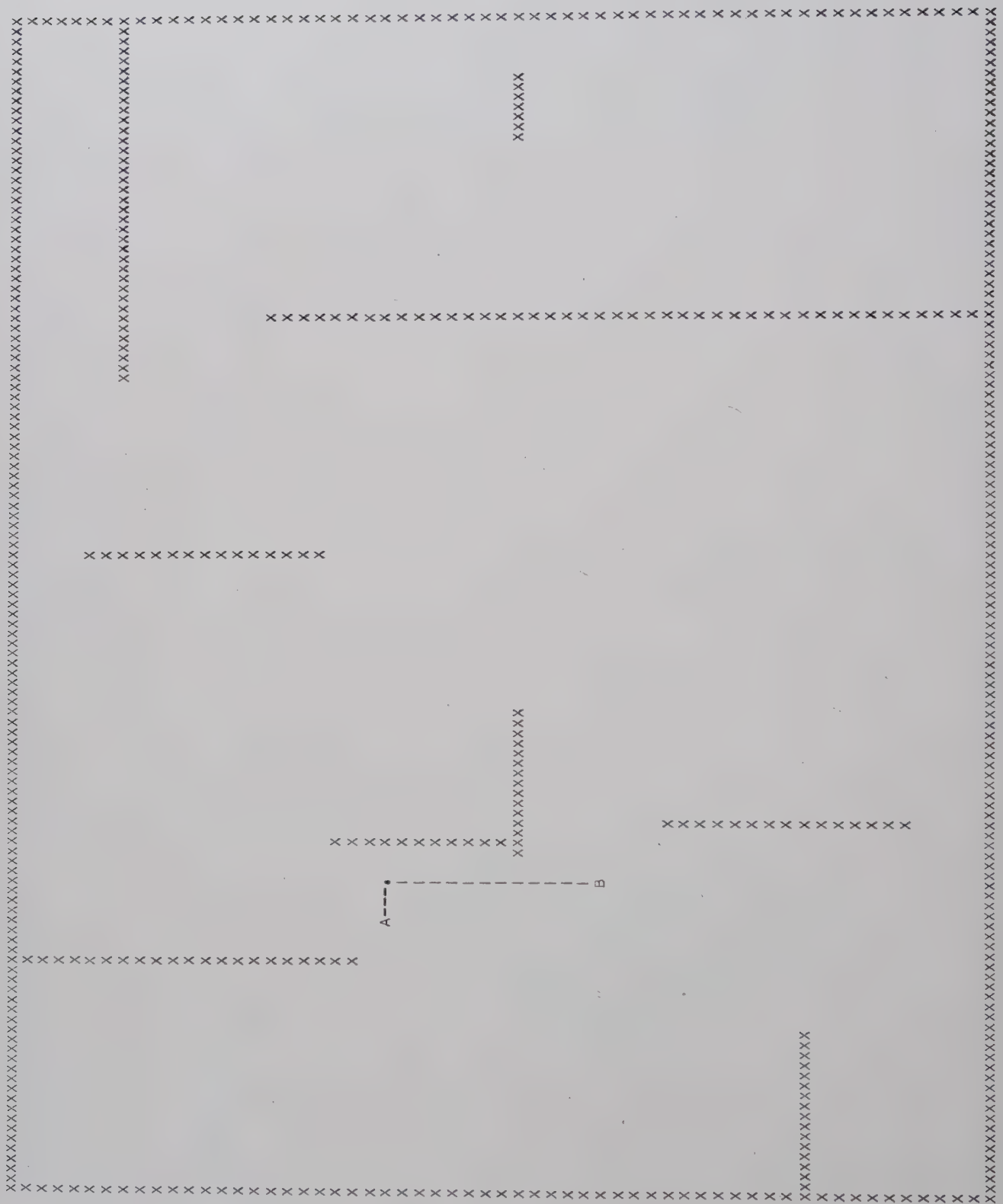


Fig. 9—One solution to the minimal-distance problem.

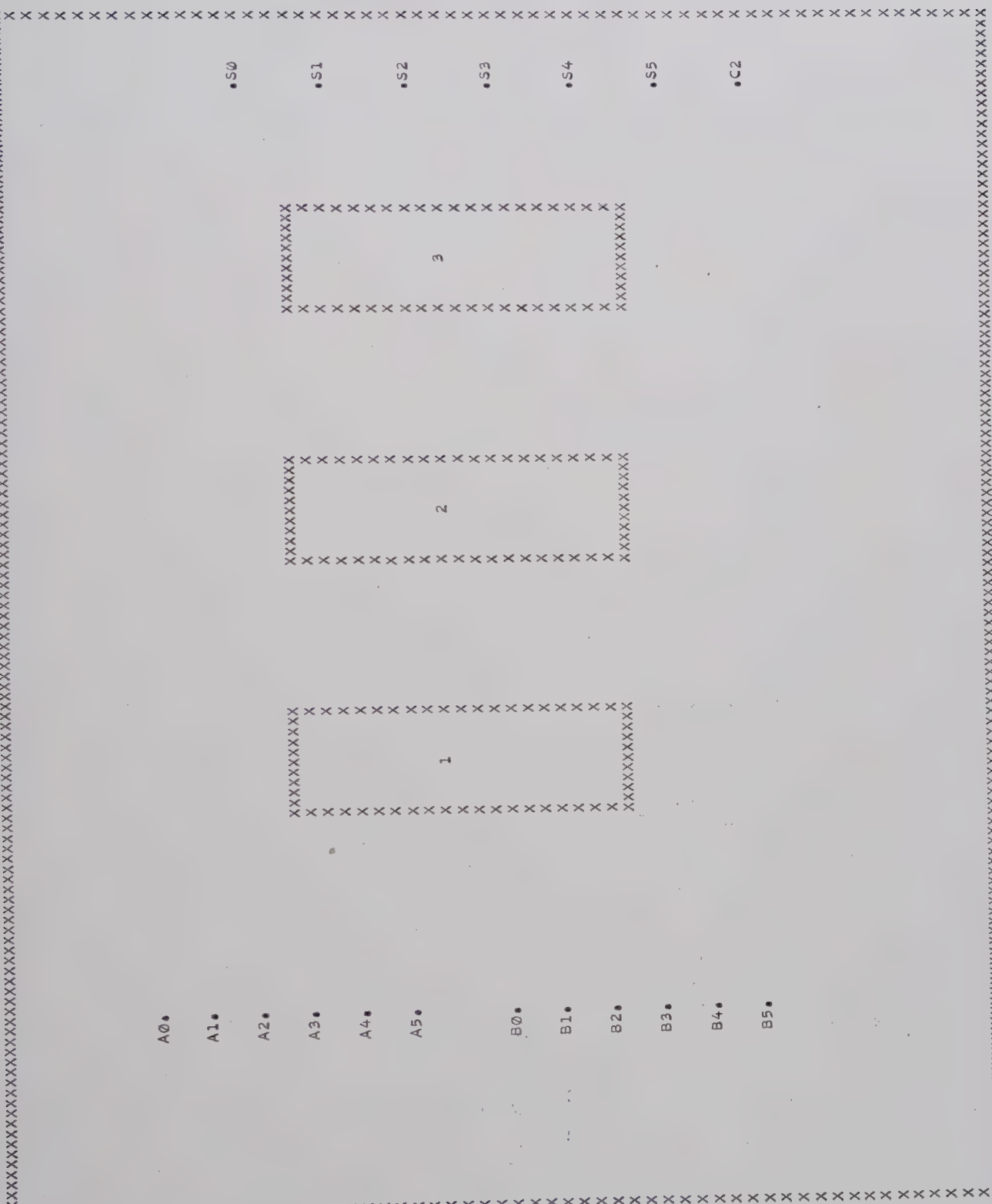


Fig. 10—A circuit block diagram as input into computer.

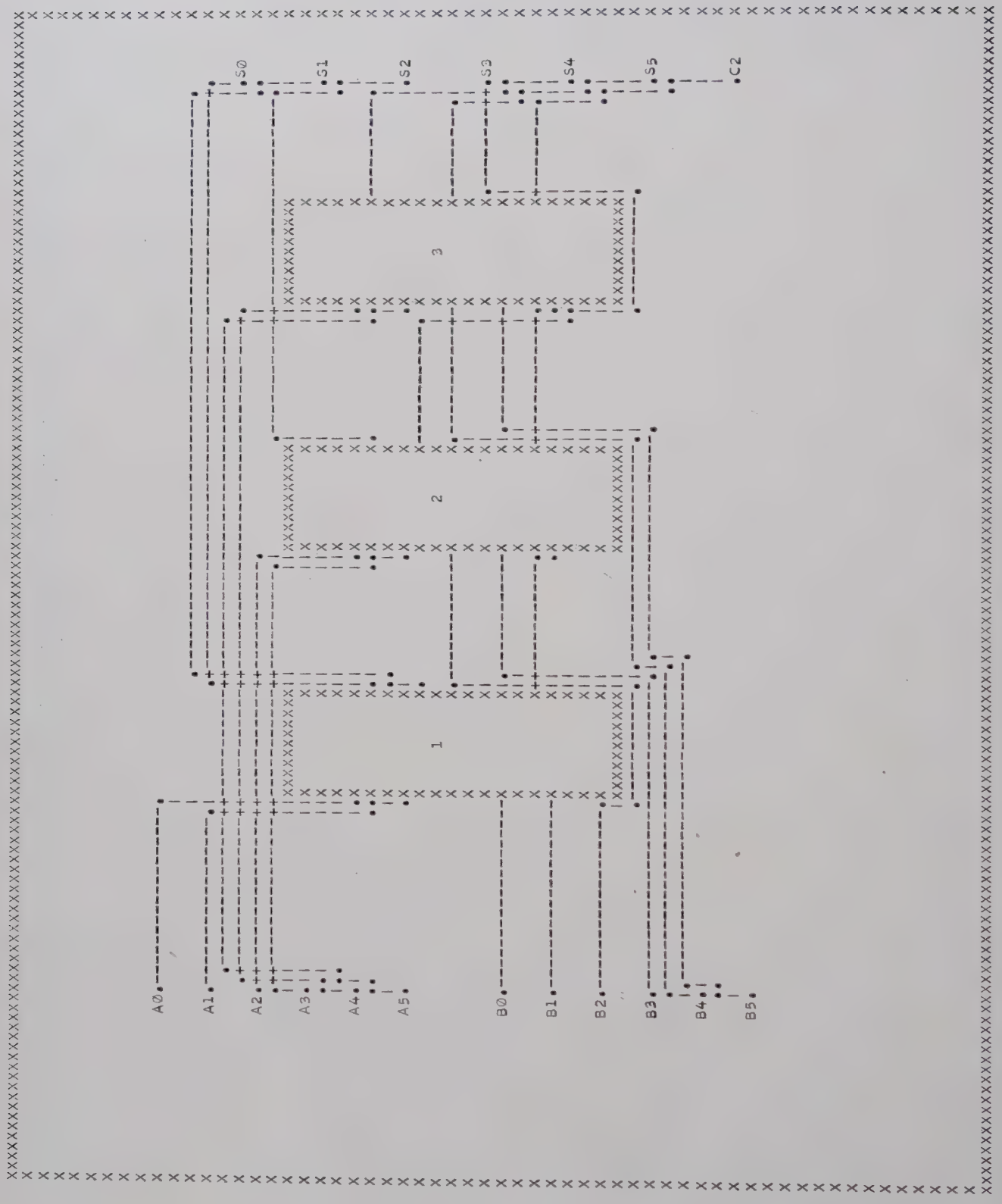


Fig. 11—Result of applying algorithm A to Fig. 10.

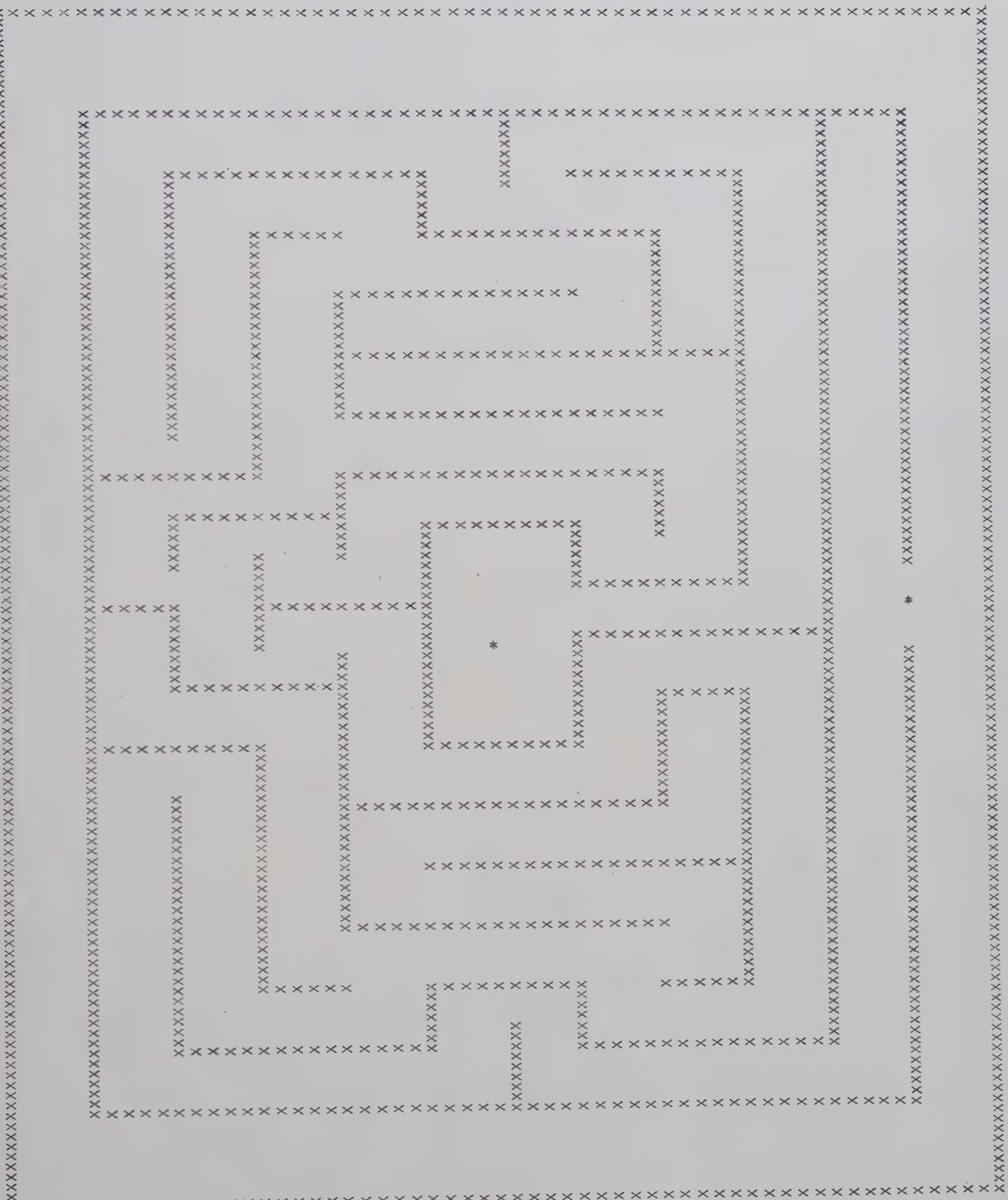


Fig. 12—A maze input into the computer.

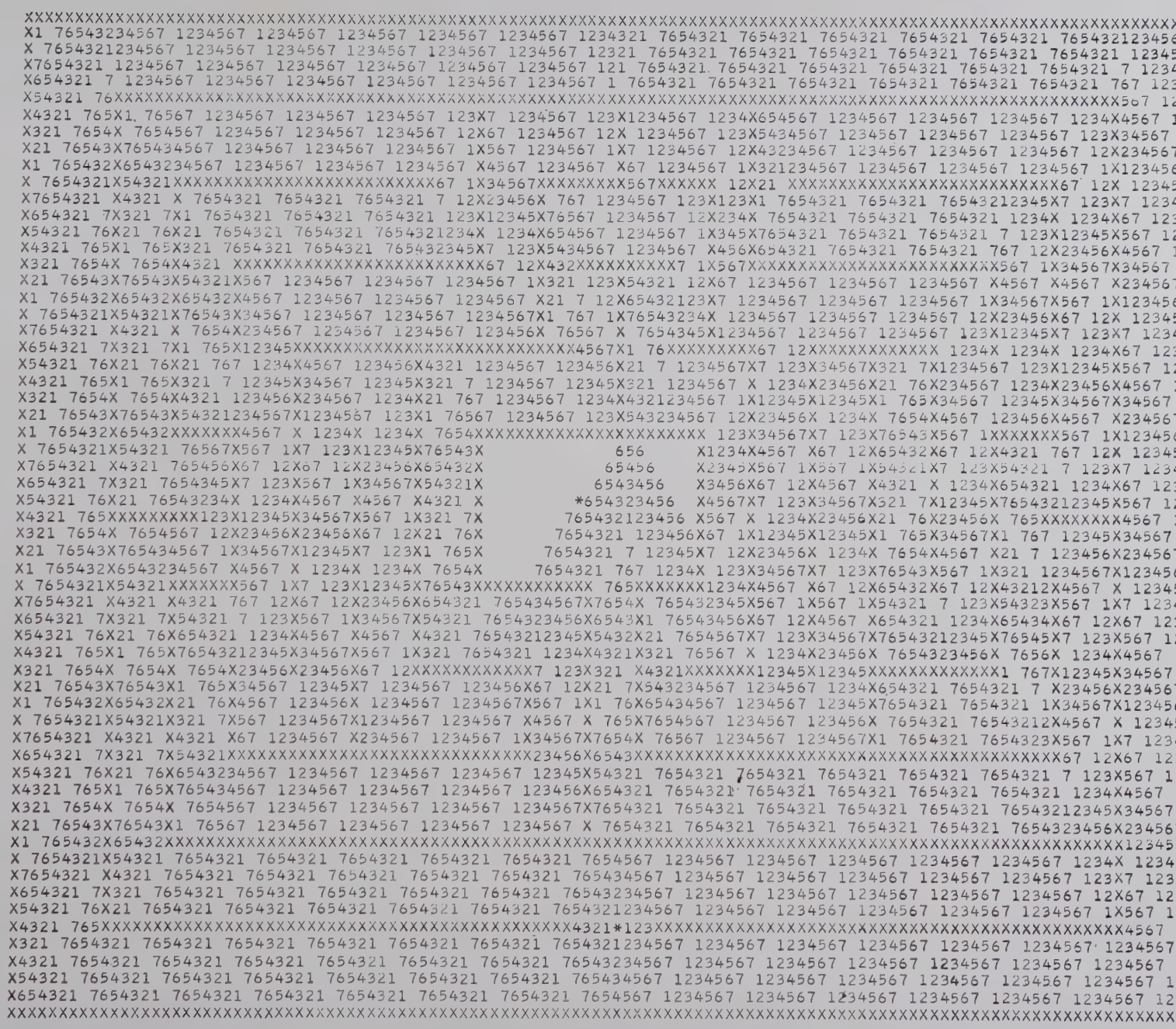


Fig. 13—Result of applying search algorithm to Fig. 12.

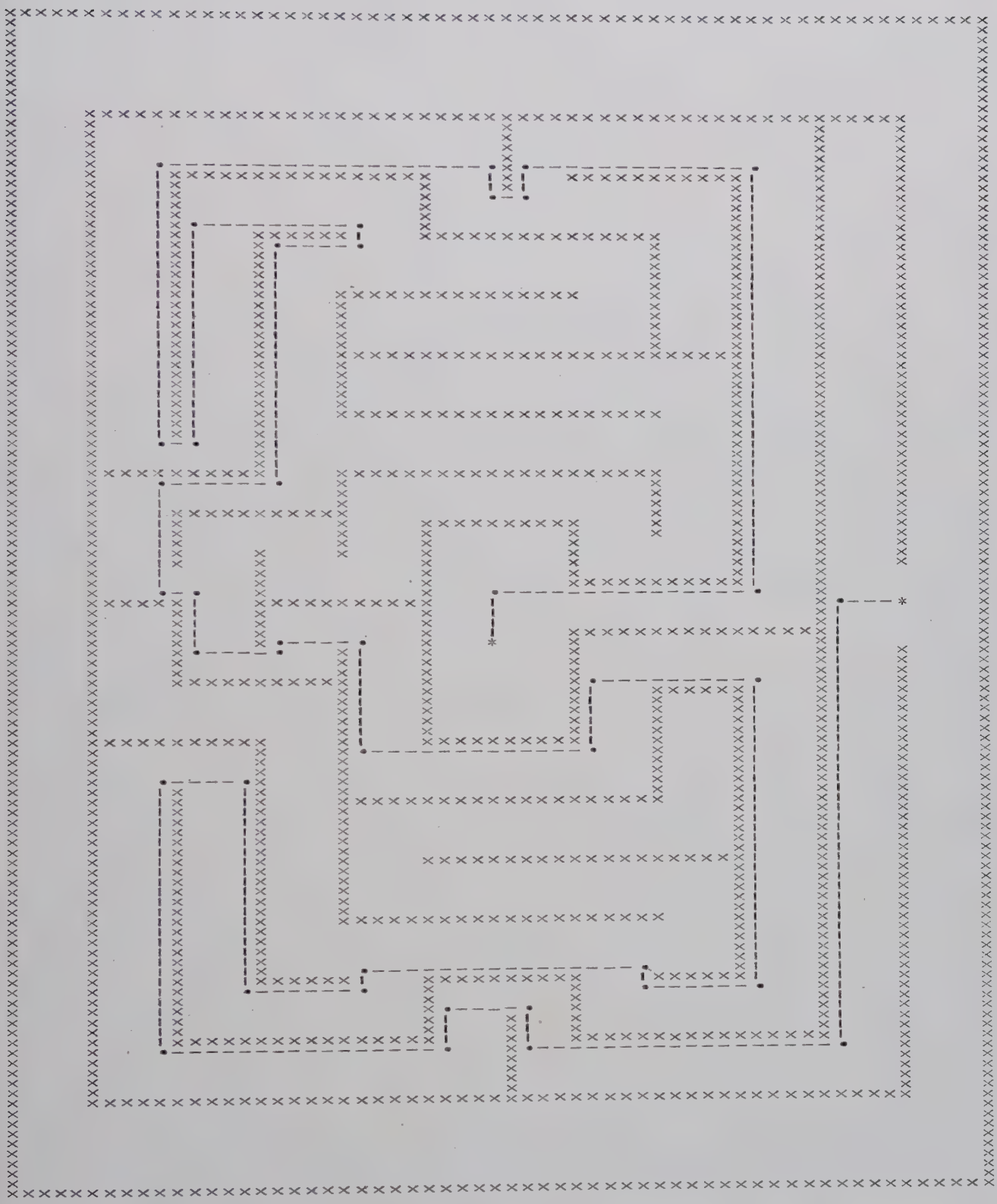


FIG. 14—Machine's solution to the maze problem.

Fig. 15—A single-slit experiment.

Fig. 16—A double-slit experiment.

Fig. 17—Beginning of a multiple-slit experiment.

Fig. 18—Near the end of a multiple-slit experiment.

Cascaded Finite-State Machines*

ARTHUR GILL†

Summary—In this paper, *networks* of finite-state machines, rather than individual machines, are discussed. The investigation centers around *cascade* networks, where the output of one machine serves as an input to another. It is shown how, by means of connection matrices, the characteristics of such a network can be obtained from those of the component machines, and how a specified machine can be decomposed into a number of cascaded components. The advantages of such a decomposition, as well as some of the problems that remain to be solved in this area, are discussed.

Introduction

MOST investigators in the field of finite-state automata[1–4] have thus far concentrated their attention on the characterization and design of *single* machines, representable by two-port "black boxes" such as those shown in Fig. 1. Thus far, only Simon[5] has investigated interconnections of such boxes, *i.e.*, *networks* of finite-state machines. The purpose of this paper is to extend such an investigation by deriving the properties of *cascaded* finite-state machines, where the output of one machine serves as an input to the next. It will be shown how the characteristics of the cascade network can be derived from the characteristics of the component machines, and how a specified machine can be decomposed into a number of cascaded components. Such a decomposition, when realizable, is quite advantageous, inasmuch as it simplifies analysis, design, installation, trouble-shooting and replacement procedures for the specified machine.

Cascaded Finite-State Machines

A finite-state machine is describable by

$$y_t = f(s_t, x_t) \tag{1}$$

$$s_{t+1} = g(s_t, x_t) \tag{2}$$

where x_t, y_t and s_t are, respectively, the input symbol, output symbol and internal state of the machine at time t. The size of the input alphabet, the size of the output alphabet and the number of states are assumed to be finite.

* Received by the PGEC, September 30, 1960; revised manuscript received, April 10, 1961. The research in this paper was supported by the U. S. Navy under contract with the University of California.

† Dept. of Elec. Engrg., University of California, Berkeley, Calif.

[1] D. A. Huffman, "The synthesis of sequential switching circuits," *J. Franklin Inst.*, vol. 257, pp. 161–190, 275–303; March–April, 1954.

[2] E. F. Moore, "Gedanken-experiments on sequential machines," in "Automata Studies," Princeton University Press, Princeton, N. J., pp. 129–153; 1956.

[3] G. H. Mealy, "Method for synthesizing sequential circuits," *Bell Sys. Tech. J.*, vol. 34, pp. 1054–1079; September, 1955.

[4] D. D. Aufenkamp and F. E. Hohn, "Analysis of sequential machines," IRE Trans. on Electronic Computers, vol. EC-6, pp. 276–285; December, 1957.

[5] J. M. Simon, "Some aspects of the network analysis of sequence transducers," *J. Franklin Inst.*, vol. 265, pp. 439–450; June, 1958.

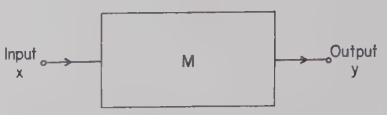


Fig. 1—A finite-state machine M.

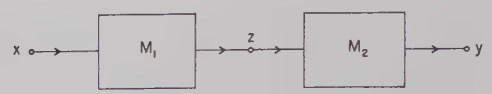


Fig. 2—Two cascaded machines.

As the simplest kind of a cascaded network, consider the two machines M_1 and M_2, shown in Fig. 2. In what follows, M_{i+1} will always stand for the machine which accepts its input from M_i. Such an interconnection implies that the output alphabet of M_i is contained within the input alphabet of M_{i+1}, which will be tacitly assumed from now on.

If the states of M_1 and M_2 are denoted by s' and s'', respectively, we have

$$z_t = f_1(s_t', x_t)$$

$$s_{t+1}' = g_1(s_t', x_t)$$

and

$$y_t = f_2(s_t'', z_t)$$

$$s_{t+1}'' = g_2(s_t'', z_t).$$

Hence,

$$y_t = f_2[s_t'', f_1(s_t', x_t)]$$

$$= f(s_t', s_t'', x_t).$$

If every ordered pair of states $\{s_t', s_t''\}$ is uniquely associated with a single state s_t, we have

$$y_t = f(s_t, x_t). \tag{3}$$

Now

$$\begin{aligned} s_{t+1} &= \{s_{t+1}', s_{t+1}''\} \\ &= \{g_1(s_t', x_t), g_2(s_t'', z_t)\} \\ &= \{g_1(s_t', x_t), g_2[s_t'', f_1(s_t', x_t)]\} \\ &= g(s_t', s_t'', x_t) \end{aligned}$$

or

$$s_{t+1} = g(s_t, x_t). \tag{4}$$

Comparing (3) and (4) with (1) and (2), it can be concluded that a cascaded network of two finite-state machines is itself a finite-state machine. The preceding derivation shows that if M_1 has n_1 states and M_2 has n_2 states, then the over-all machine has $n_1 n_2$ states. By

simple induction, it can be concluded that a cascade network of N machines $M_1, M_2, \cdots, M_N$ with $n_1, n_2, \cdots, n_N$ states, respectively, is equivalent to a single machine with $n_1 n_2 \cdots n_N$ states. Such a network is shown in Fig. 3.

If the over-all machine, denoted by M, is a reduced machine,[2] then every component M_i must also be a reduced machine. To prove, suppose M_1 is reducible from n_1 to $\bar{n}_1$ states ($\bar{n}_1 < n_1$); hence, M is reducible from $n_1 n_2 \cdots n_N$ to $\bar{n}_1 n_2 \cdots n_N$ states, which contradicts the assumption that M is in a reduced form.

From the definition of strong-connectedness[2] and the manner in which the states of M are defined, it can also be established that if any of the M_i is not strongly connected, then M cannot be a strongly connected machine.

Connection Matrices

The characterizing equations (1) and (2) can be expressed schematically by means of a "state diagram." A branch in the diagram which points from vertex i to vertex j and is labeled (ξ/η), indicates that the input symbol ξ takes state i into state j, yielding the output symbol η. If transition from i to j is caused by more than one input symbol, the corresponding branch is labeled $(\xi_1/\eta_1)+(\xi_2/\eta_2)+\cdots+(\xi_r/\eta_r)$, where the (ξ_i/η_i) are the "input-output pairs" causing the transition. Fig. 4 shows the state diagram of a 3-state machine M_1 and a 2-state machine M_2, both machines having binary input and output alphabets.

An alternative way of describing a machine is through a connection matrix.[4] For an n-state machine, this matrix is an $n \times n$ array, where the element common to row i and column j is precisely the label attached to the branch pointing from vertex i to vertex j in the state diagram. The absence of a branch is designated by 0. The connection matrix for a machine M will be designated by $[M]$. Thus, for the machines of Fig. 4, we have

$$[M_1] = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} (1/1) & (0/1) & 0 \\ (0/1) & 0 & (1/0) \\ (0/1)+(1/1) & 0 & 0 \end{bmatrix} \end{array} \tag{5}$$

$$[M_2] = \begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} (1/0) & (0/1) \\ (0/1) & (1/1) \end{bmatrix} \end{array}. \tag{6}$$

For a deterministic machine without any input restrictions (*i.e.*, for which every input symbol is permissible, regardless of the state), every row in the matrix contains exactly m input-output pairs, where m is the size of the input alphabet; all input symbols in any given row are distinct.

Since the cascade network M of two finite-state machines M_1 and M_2 is itself a finite-state machine, it is similarly describable by a connection matrix. The

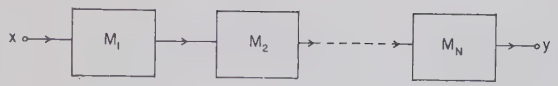


Fig. 3—N cascaded machines.

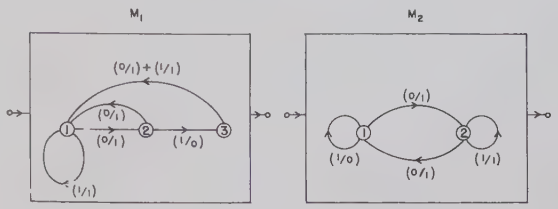


Fig. 4—Two binary machines.

states of M, as defined in the previous section, are in one-to-one correspondence with pairs of states picked up from M_1 and M_2. It is convenient to denote the state of M which corresponds to states i of M_1 and k of M_2 by i-k. Thus, in the example of Fig. 4, the machine produced by connecting the output of M_1 to the input of M_2 has the states 1-1, 1-2, 2-1, 2-2, 3-1, 3-2.

For the purpose of constructing the connection matrix $[M]$ from $[M_1]$ and $[M_2]$, it is convenient to define the following "multiplication" rule: if (ξ_1/η_1) and (ξ_2/η_2) are two input-output pairs, then

$$(\xi_1/\eta_1)(\xi_2/\eta_2) = \begin{cases} (\xi_1/\eta_2) & \text{if } \eta_1 = \xi_2 \\ 0 & \text{if } \eta_1 \neq \xi_2. \end{cases} \tag{7}$$

(When $\eta_1 = \xi_2$, this operation may be viewed as a conventional multiplication, where η_1 and ξ_2 are canceled from the product.) In addition, the following rules are applicable: if A, B and C are input-output pairs, then

$$A + B = B + A$$
$$A(B + C) = AB + AC$$
$$A + 0 = A$$
$$A0 = 0A = 0$$

where all multiplications are carried out as defined by (7).

In terms of the above operations, the elements m_{ij} of $[M]$ can be readily related to the elements m_{ij}' of $[M_1]$ and m_{ij}'' of $[M_2]$ as follows:

$$m_{i-k,j-l} = m_{ij}' m_{kl}''. \tag{8}$$

Eq. (8) can be verified by noticing the following: if M_1 is in state i and M_2 in state k, then M is in state i-k; transition of M into state j-l is possible only if the output symbol in m_{ij}' and the input symbol in m_{kl}'' are identical, in which case the input symbol in $m_{i-k,j-l}$ is the same as that in m_{ij}', and the output symbol in $m_{i-k,j-l}$ is the same as that in m_{kl}''.

The matrix $[M]$ will be referred to as the "product" of $[M_1]$ and $[M_2]$, and designated by $[M_1][M_2]$. Clearly, since $m_{ij}' m_{kl}'' \neq m_{kl}'' m_{ij}'$, in general $[M_1][M_2] \neq [M_2][M_1]$. If M is the cascade connection of M_1, M_2

and M_3, we have

$$[M] = [M_1]\{[M_2][M_3]\} = \{[M_1][M_2]\}[M_3]$$
$$= [M_1][M_2][M_3].$$

In general: if M is the cascade connection of M_1, M_2, $\cdots$, M_N, then

$$[M] = [M_1][M_2] \cdots [M_N].$$

For the example of Fig. 4, using (5) and (6), we have

$$[M] = [M_1][M_2] = \begin{array}{c} \\ 1\text{-}1 \\ 1\text{-}2 \\ 2\text{-}1 \\ 2\text{-}2 \\ 3\text{-}1 \\ 3\text{-}2 \end{array} \begin{array}{c} \begin{array}{cccccc} 1\text{-}1 & 1\text{-}2 & 2\text{-}1 & 2\text{-}2 & 3\text{-}1 & 3\text{-}2 \end{array} \\ \begin{bmatrix} (1/0) & 0 & (0/0) & 0 & 0 & 0 \\ 0 & (1/1) & 0 & (0/1) & 0 & 0 \\ (0/0) & 0 & 0 & 0 & 0 & (1/1) \\ 0 & (0/1) & 0 & 0 & (1/1) & 0 \\ (0/0)+(1/0) & 0 & 0 & 0 & 0 & 0 \\ 0 & (0/1)+(1/1) & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}.$$

As an example, the element $m_{3\text{-}2,1\text{-}2}$ is determined as follows:

$$m_{3\text{-}2,1\text{-}2} = m_{3,1}'m_{2,2}''$$
$$= [(0/1) + (1/1)](1/1)$$
$$= (0/1)(1/1) + (1/1)(1/1)$$
$$= (0/1) + (1/1).$$

The state diagram of M is shown in Fig. 5.

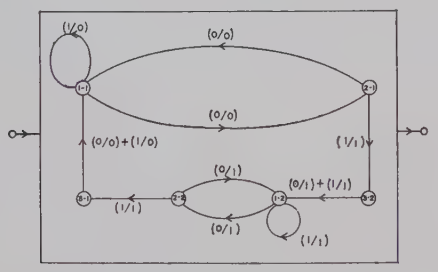


Fig. 5—The cascade combination of M_1 and M_2.

Decomposability of Finite-State Machines

The construction of a finite-state machine as a cascade network of finite-state "stages" is in many ways more desirable than the construction of a single unit. The reason is that a cascade construction enables one to separate readily a portion of the machine from the rest, thus facilitating analysis and design operations, as well as installation, testing and repairing routines. It is, therefore, of interest to examine the conditions under which a specified machine is decomposable into a number of cascaded components.

The following discussion will be restricted to the problem of "simple decomposability," *i.e.*, decomposability into exactly two stages. Clearly, if a procedure is available for determining all simple decompositions of a given machine, the same procedure, when applied successively to each component, can serve to determine all possible decompositions, simple and otherwise.

The first decomposability condition is obvious: a machine is decomposable into a cascade network of an n_1-state machine and an n_2-state machine only if it is equivalent to an n_1n_2-state machine. If the number of states in the given machine is not the desired product, redundant states may always be added to force this condition. It is also clear that every given machine can be trivially decomposed by inserting at its output the "identity machine" whose connection matrix is

$$[I] = [(\eta_1/\eta_1) + (\eta_2/\eta_2) + \cdots + (\eta_p/\eta_p)]$$

where $\eta_1, \eta_2, \cdots, \eta_p$ is the output alphabet of the given machine. Thus, for every machine M,

$$[M] = [M][I].$$

Or, if $\eta_1, \eta_2, \cdots, \eta_p$ is the output alphabet of M:

$$[M] = [I][M].$$

Decomposition, of course, is beneficial only if each component is simpler than the given machine, or when

$$n_1 + n_2 \leq n_1n_2,$$

which is true only when both n_1 and n_2 exceed unity. In the example of Figs. 4 and 5 the over-all machine has 6 states, while the components have 3 states and 2 states, respectively. Thus, handling any of the components is simpler than handling the over-all machine (although the over-all memory requirements are now greater).

At this point it is useful to introduce the following definition: an $n_1n_2 \times n_1n_2$ matrix is "$n_1 - n_2$ partitioned" when it is partitioned into n_1^2 $n_2 \times n_2$ submatrices. As an example, consider the following 6×6 connection matrix, which corresponds to the state diagram of Fig. 6:

$$[M] = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \left[\begin{array}{ccc|ccc} 0 & (1/1) & 0 & (0/1) & 0 & 0 \\ (1/1) & 0 & 0 & 0 & 0 & (0/0) \\ (1/1) & 0 & 0 & (0/1) & 0 & 0 \\ \hline (0/1) & 0 & 0 & (1/1) & 0 & 0 \\ 0 & 0 & (0/0) & 0 & 0 & (1/0) \\ (0/1) & 0 & 0 & (1/1) & 0 & 0 \end{array}\right] \end{array}. \quad (9)$$

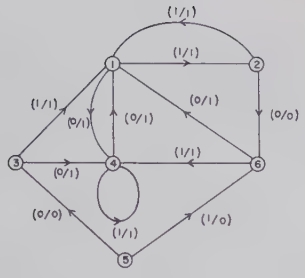


Fig. 6—A 6-state machine.

As shown, the matrix is 2-3 partitioned. Since state labels are arbitrary, a corresponding permutation of rows and columns yields an identical machine; consequently, an n_1-n_2 partitioning of a matrix is not unique and may have as many as $(n_1n_2)!$ different versions.

Now, a necessary condition for an n_1n_2-state machine M to be decomposable into an n_1-state machine M_1 and an n_2-state machine M_2 is the following: there must exist an n_1-n_2 partition of $[M]$, such that in any given submatrix each row contains the same set of input symbols. A matrix which fulfills this condition will be referred to as "potentially decomposable." Matrix (9) above is clearly potentially decomposable.

To prove the above condition, recall that each state of M is uniquely associated with one state of M_1 and one state of M_2. Thus, states 1, 2, $\cdots$, n_1n_2 of M could be associated with the state-pairs 1-1, 1-2, $\cdots$, $1-n_2$, 2-1, 2-2, $\cdots$, 2-n_2, $\cdots$, n_1-1, n_1-2, $\cdots$, n_1-n_2 of M_1 and M_2. Carrying out this association in matrix (9), we have

$$[M] = \begin{array}{c|cccc|cc} & 1\text{-}1 & 1\text{-}2 & 1\text{-}3 & 2\text{-}1 & 2\text{-}2 & 2\text{-}3 \\ \hline 1\text{-}1 & 0 & (1/1) & 0 & (0/1) & 0 & 0 \\ 1\text{-}2 & (1/1) & 0 & 0 & 0 & 0 & (0/0) \\ 1\text{-}3 & (1/1) & 0 & 0 & (0/1) & 0 & 0 \\ \hline 2\text{-}1 & (0/1) & 0 & 0 & (1/1) & 0 & 0 \\ 2\text{-}2 & 0 & 0 & (0/0) & 0 & 0 & (1/0) \\ 2\text{-}3 & (0/1) & 0 & 0 & (1/1) & 0 & 0 \end{array} . \quad (10)$$

It can now be seen that each submatrix in the partition of $[M]$ corresponds to a transition between a pair of states in M_1, and each row within a submatrix corresponds to a different state in M_2. Since transition between two states in M_1 is caused by the same set of input symbols regardless of the state of M_2, and since the input of M_1 is also the input to the over-all machine M, the condition follows.

If M is potentially decomposable, this condition would be revealed by $n_1!n_2!$ different partitions. For small matrices, the property can be discovered or found to be absent by inspection. For large matrices, a computer may be employed to automatically permute the rows and columns until the condition is fulfilled, or until the condition is found to be absent in $(n_1n_2)!-n_1!n_2!+1$ partitions. If the machine is found to be potentially decomposable, decomposability can be determined in a constructive manner, as will be shown in the next section.

Machine Decomposition

The decomposition of a machine whose connection matrix is given, is best demonstrated by means of an example. Table I is the "decomposition table" for the machine specified by matrix (10). Desired is a decomposition of M into two binary machines—M_1 with 2 states and M_2 with 3 states. In the table, the labels attached to the rows refer to elements in $[M_1]$, and labels attached to columns refer to elements in $[M_2]$; (i, j) stands for the element in $[M_1]$ or $[M_2]$ which is common to row i and column j in the corresponding matrix. The elements in the table are ordered in such a manner that each row in $[M_1]$ or in $[M_2]$ forms a "subtable" (shown bordered by a bold-face line). The table is completed in the following manner: if the element $m_{i\text{-}k,j\text{-}l}$ in $[M]$ is nonzero, it is copied onto the cell common to row (i, j) and column (k, l) in the table; otherwise the cell is left blank. The circled numbers serve merely as an aid to the reader in the ensuing discussion.

The column labeled $[M_1]$ and row labeled $[M_2]$ contain the elements assigned to the matrices of the component machines M_1 and M_2, respectively. These elements are determined as follows: since ①, given as (1/1), must be the product of ④ and ⑱, ④ must be

TABLE I

Decomposition Table for Machine Specified by Matrix (10)

M_1 \ M_2	(1/1)	(1, 2)	(1, 3)	(2, 1)	(2, 2)	(2, 3)	(3, 1)	(3, 2)	(3, 3)	$[M_1]$
(1, 1)		(1/1) ①		(1/1) ②			(1/1) ③			(1/α) ④
(1, 2)	(0/1) ⑤					(0/0) ⑥	(0/1) ⑦			(0/β) ⑧
(2, 1)	(0/1) ⑨					(0/0) ⑩	(0/1) ⑪			(0/β) ⑫
(2, 2)	(1/1) ⑬					(1/0) ⑭	(1/1) ⑮			(1/β) ⑯
$[M_2]$	(β/1) ⑰	(α/1) ⑱		(α/1) ⑲		(β/0) ⑳	(α/1) ㉑ (β/1) ㉒			

$(1/\alpha)$ and ⑱ must be $(\alpha/1)$, where α is an arbitrary symbol. Since ⑤, given as $(0/1)$, must be the product of ⑧ and ⑰, ⑧ must be $(0/\beta)$ and ⑰ must be $(\beta/1)$; as ⑰ and ⑱ appear in the same row of $[M_2]$, β must differ from α. Similarly, ② implies ⑲, ⑥ implies ⑧ and ⑳, ③ implies ㉑, ⑪ implies ⑫ and ㉒, and ⑬ and ⑰ imply ⑯ (in that order). It is now necessary to check the table for "compatibility," *i.e.*, to ascertain that every entry in the table is the product of the $[M_1]$ and $[M_2]$ entries in its column and row (blank cells are equivalent to 0). For example: ⑭ must be the product of ⑯ and ⑳. If the table is not compatible, the corresponding partition of $[M]$ is inadequate and another partition should be tried. As can be easily verified, Table I is compatible, and hence M_1 and M_2, as determined in the table, are the desired components

$$[M_1] = \begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} (1/\alpha) & (0/\beta) \\ (0/\beta) & (1/\beta) \end{bmatrix} \end{array} \tag{11}$$

$$[M_2] = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} (\beta/1) & (\alpha/1) & 0 \\ (\alpha/1) & 0 & (\beta/0) \\ (\alpha/1)+(\beta/1) & 0 & 0 \end{bmatrix} \end{array}. \tag{12}$$

If α is taken as "0" and β as "1," (11) and (12) are seen to be identical with matrices (6) and (5), respectively; thus, the machine of Fig. 6 can be constructed by connecting M_1 and M_2 of Fig. 4 in a reversed order.

It is seen that the determination of $[M_1]$ and $[M_2]$ via the decomposition table is governed by the following requirements: 1) every entry must be the product of the $[M_1]$ and $[M_2]$ entries in its column and row; and 2) within each subtable, every input symbol must be represented in column $[M_1]$ and row $[M_2]$ exactly once (the requirement for $[M_1]$ is already fulfilled when M is potentially decomposable). As these requirements are necessary and sufficient for the decomposition of a given partitioned matrix, the failure to satisfy them implies that the given partition should be discarded. When these requirements are satisfiable, they can be used to successively complete the $[M_1]$ and $[M_2]$ entries and hence to yield the desired components. The resulting decomposition is, in general, not unique; the "intermediate alphabet" (the output alphabet of M_1 and input alphabet of M_2) may be arbitrarily selected.

Conclusions

The present paper is an initial attempt to consider networks of finite-state machines, rather than individual ones. Treated was the most obvious type of network—a cascade connection of finite-state stages. The ideas and techniques developed in the paper may be applied directly to interconnections of machines where each unit accepts its input from one unit only, but may feed its output to any number of units. An example is shown in Fig. 7. A special case of a cascade network is an autonomous two-stage network as shown in Fig. 8.

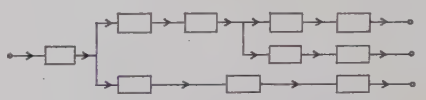


Fig. 7—A finite-state network.

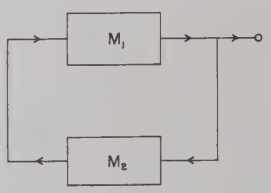


Fig. 8—An autonomous two-stage network.

An interesting problem which arises in connection with this network is the following: given the machine M_1, construct an auxiliary machine M_2 (a "controller") that would take M_1 automatically into some specified state. The solution to this problem is greatly facilitated by the techniques introduced in this article, and will appear in a later paper.

The synthesis of cascade networks, as treated in this paper, represents a preliminary investigation and undoubtedly can stand improvement. The "potential decomposability" test is tedious, and calls for an algorithm which will reveal this property without an exhaustive search. In addition, an algorithmic procedure is desired for the actual decomposition operation, which will facilitate the processing of large decomposition tables by automatic means. Another important problem, that of rendering a given machine decomposable by the addition of redundant states, was found to be quite difficult, and thus far has not been satisfactorily solved.

The Realization of Symmetric Switching Functions with Linear-Input Logical Elements*

WILLIAM H. KAUTZ†, MEMBER, IRE

Summary—**The problem of synthesizing switching networks out of linear-input (threshold) elements is studied for the class of symmetric switching functions. Tight bounds are derived for the number of elements required in a minimal realization, and a method of synthesis is presented which yields economical networks. Minimal networks result for all symmetric functions of no more than about twelve variables, and for several other cases. In particular, it is shown how the parity function of any number n of variables can be realized with about $\log_2(n)$ elements.**

AN outstanding problem in combinational switching theory concerns the realization of an arbitrary switching function f of n variables $x_1, x_2, \cdots, x_n$ in a network of logical elements, each of which is describable by a *linear-input* function,[1] also called setting function,[2] threshold function,[2] or linearly separable function.[3] A linear-input function is a switching function $h(y_1, y_2, \cdots, y_m)$ which takes on the value 1 or 0 in accordance with whether the linear equality

$$\alpha_1 y_1 + \alpha_2 y_2 + \cdots + \alpha_m y_m \geq \alpha_0$$

is or is not satisfied. The *weights* α_i and *threshold* α_0 are real, positive or negative constants, which may be taken to be integers without loss of generality. A reasonable symbol for this element is shown in Fig. 1. Several simple linear-input devices have been conceived which show promise of leading to economical digital networks for the realization of arbitrary logical operations.[1]

At present, there exists no satisfactory analytical framework for the analysis of this class of networks, nor are there known any procedures for economical synthesis.

This paper concerns the realization of the class of *symmetric* switching functions with networks of linear-input elements. We present an approach to this synthesis problem which yields 1) tight bounds on the minimum number of such elements required to realize any symmetric function, 2) an analytical viewpoint which permits the synthesis of any symmetric function of up to about 12 variables, and several other cases, using the minimal number of elements, and 3) a minimal realization of the alternating symmetric or *parity* function

$$f_p = x_1 \oplus x_2 \oplus \cdots \oplus x_n$$

which requires only $1+[\log_2(n)]$ elements. (The symbol $\oplus$ indicates exclusive-OR; the brackets denote the integer part of the quantity within.)

Previously published solutions for the case of the parity function have required from $1+[n/2]$ to $1+n$ elements.[1,4] The improved solution presented below makes possible simple parity-checking and error-correction circuits for applications to the recording and transmission of digital data. A 7-input parity gate, for example, requires only 3 linear-input elements (Fig. 2); a gate with up to 15 inputs requires 4 elements, etc.

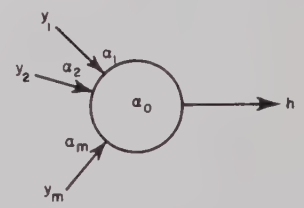


Fig. 1—Symbol for the linear-input element.

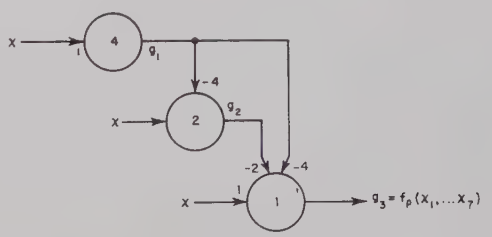


Fig. 2—Minimal linear-input network for the 7-variable parity function.

The Parity Function

Consider first the three-element network shown in Fig. 2. In this and subsequent figures, the generic input labeled x symbolically designates the entire set of inputs $x_1, x_2, \cdots, x_n$, each with unit weight. The variable x takes on the values 0, 1, 2, $\cdots$, n, to indicate the number of x_i which have the value 1. It is well known that any symmetric function f_s of n (uncomplemented) variables may be described by a listing of these x values

* Received by the PGEC, January 13, 1961; revised manuscript received, May 9, 1961.

† Stanford Research Institute, Menlo Park, Calif.

[1] R. C. Minnick, "Linear-input logic," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-10, pp. 6–16; March, 1961.

[2] M. C. Paull and E. J. McCluskey, Jr., "Boolean functions realizable with single threshold devices," PROC. IRE, vol. 48, pp. 1334–1337; July, 1960.

[3] R. McNaughton, "Unate truth functions," Appl. Math, and Statistical Lab., Stanford University, Stanford, Calif., October, 1957; IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-10, pp. 1–6, March, 1961.

[4] S. Muroga, "Logical elements on majority decision principle and complexity of their circuit," *Proc. Internatl. Conf. on Information Processing*, Paris, France, June, 1959, UNESCO House, Paris, pp. 400–407; 1960.

for which $f_s=1$; *e.g.*, in Shannon's notation,[5]

$$f_p(x_1, \cdots, x_7) = S_{1,3,5,7}(x_1, \cdots, x_7),$$

the x values subscripted to S indicate that the parity function of seven variables equals 1 when and only when an *odd* number of x_i equal 1.

The three element outputs of Fig. 2 are shown plotted against x in Fig. 3. The circuit operates as follows. The first element generates g_1, which equals 0 until x is increased to the threshold 4, and equals 1 for $x \geq 4$. The second element behaves similarly with threshold 2 so long as $x<4$, since $g_1=0$; when $x \geq 4$, however, the presence of $g_1=1$ increases the apparent threshold of the second element to $2+4=6$. Thus g_2 has two positive transitions in value instead of one. The number of positive transitions is redoubled in g_3: the third element operates with threshold 1 when $g_1=g_2=0$, but the apparent threshold increases to 3 when $g_1=0$, $g_2=1$; to 5 when $g_1=1$, $g_2=0$; and finally to 7 when $g_1=g_2=1$. Thus, the circuit of Fig. 2 realizes the parity function $f_p=g_3$ of seven variables (or fewer, since $x \leq n$).

It is clear that this process may be continued to achieve 8 positive transitions at the output of a fourth element, etc., and in general 2^{r-1} positive transitions at the output of the rth element, provided only that the weights and thresholds are scaled up properly as each additional element is added. Using the notation indicated in Fig. 4(a) for a general network of this "feed-forward" type, we may apply the above analysis directly to verify that the set of values

$$\beta_{jk} = 2^{r-j}, \qquad j \leq k, j, k = 1, 2, \cdots, r,$$

gives the output function

$$f_p = S_{1,3,5,\cdots 2^r-1}(x_1, x_2, \cdots, x_{2^r-1})$$

—that is, the parity function of $n=2^r-1$ (or fewer) variables. The next lower value of r cannot be used until n is reduced to $2^{r-1}-1$, so that a number of elements equal to

$$r = 1 + [\log_2 (n)]$$

is always adequate for the parity function of n variables.

For the *even* parity function $\bar{f}_p=S_{0,2,4,\cdots}$, a network transformation proposed by Minnick allows the output of any network to be complemented by systematically changing only the weights and thresholds of the elements, and does not increase the number of elements.[1] Thus this lower bound is valid for both the odd and even alternating symmetric functions.

It is clear from the construction that in such a "feed-forward" type of network, the number of positive transitions in g_r cannot exceed 2^{r-1}, the maximum number of different ways in which the $r-1$ previous element outputs can be combined together to yield different apparent thresholds for the rth element. Thus no fewer number of elements can produce this parity function, which has n transitions, and the proposed realization is minimal in the class of "feed-forward" networks.

The class of "feed-forward" networks is that in which the elements can be numbered 1, 2, $\cdots$, r such that

[5] C. Shannon, "A symbolic analysis of relay and switching circuits," *Trans. AIEE*, vol. 57, pp. 713–723; 1938.

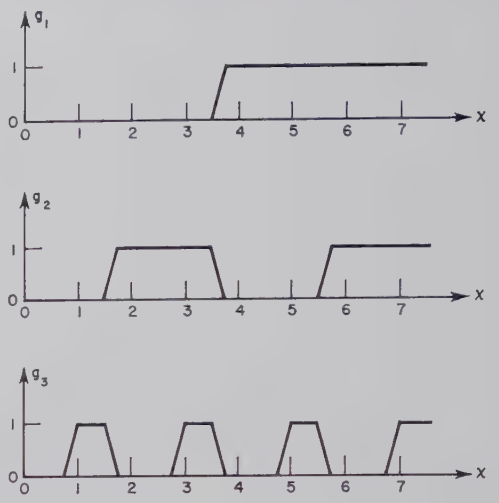


Fig. 3—Element-output functions for the network of Fig. 2

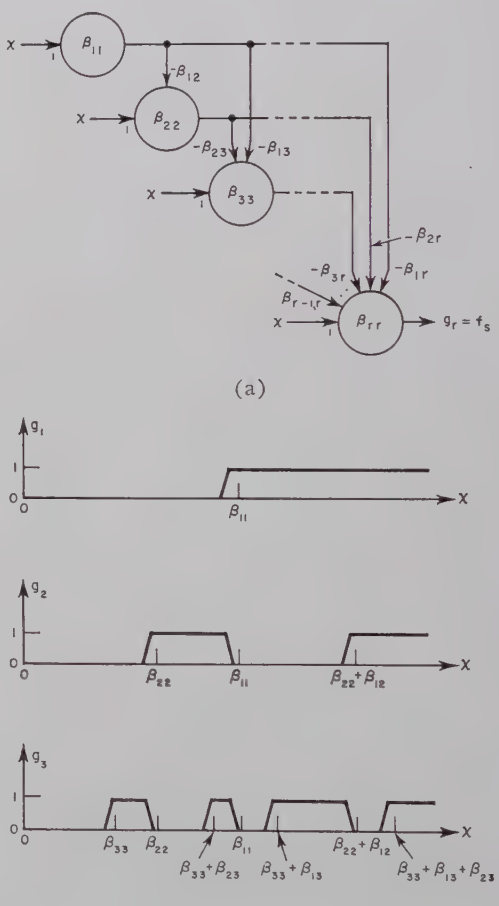


Fig. 4—General feed-forward linear-input network, and the corresponding element-output functions.

each element receives inputs only from x and from the outputs of lower-numbered elements. However, a simple argument shows that a feed-forward network is the most general form of a *loop-free* network. For, in the absence of any closed loops, a path traced backwards from the output f through non-x input arrows must eventually terminate on some element; call this element #1, momentarily delete it, and repeat to locate element #2, etc. By this construction, each element output can be connected only to higher-numbered element inputs, and the feed-forward feature follows. Thus the realization presented is minimal over the class of loop-free networks.

If loops are allowed in a network of linear-input elements, a form of memory or storage can result. This storage can be used constructively for the design of sequential networks, but can produce nonunique or oscillating outputs in a network intended for purely combinational use. This does not imply that loops are never desired, for it is conceivable that in networks of sufficient complexity, nonstorage loops must be allowed if minimality is to be achieved. Examples of this phenomenon are already known in other types of combinational switching networks.[6] However, it is not known whether or not loops are required in minimal linear-input networks.

General Analysis

Consider now the form of the element outputs $g_1, g_2, \cdots, g_r$ when the weights and thresholds are not so restricted as above. The graphs of Fig. 4(b) display the transition x values in terms of the weights β_{kj} of Fig. 4 for $r=3$. We assume for the moment an ordering of β values which allows the maximum number (namely, four) of positive transitions in g_3. It is apparent from the parity example and from this figure that, in general, 1) the x values at which positive transitions occur in g_k define negative transitions in $g_{k+1}, g_{k+2}, \cdots, g_r$, and 2) whereas these positive-transition x values are completely unconstrained for g_1 and g_2, falling at the values

$$g_1\colon \beta_{11} \qquad g_2\colon \beta_{22}, \quad \beta_{22}+\beta_{12},$$

they are somewhat restricted for g_3, g_4, etc.; *e.g.*,

$$g_3\colon \beta_{33},\quad \beta_{33}+\beta_{23},\quad \beta_{33}+\beta_{13},\quad \beta_{33}+\beta_{23}+\beta_{13}.$$

Thus, if three of these values are specified, the fourth is determined. This constraint may be conveniently expressed by noting that if these x values are written as the exponents of a polynomial in a variable z, then this polynomial may be factored, *e.g.*,

$$g_1\colon P_1(z) = z^{\beta_{11}} \qquad g_2\colon P_2(z) = z^{\beta_{22}} + z^{\beta_{22}+\beta_{12}} = z^{\beta_{22}}(1+z^{\beta_{12}})$$

and

$$g_3\colon P_3(z) = z^{\beta_{33}} + z^{\beta_{33}+\beta_{23}} + z^{\beta_{33}+\beta_{13}} + z^{\beta_{33}+\beta_{23}+\beta_{13}} = z^{\beta_{33}}(1+z^{\beta_{23}})(1+z^{\beta_{13}}).$$

In general, the r constants $\beta_{kr}(k=1, 2, \cdots, r)$ associated with element r are made manifest in the factored form of $P_r(z)$, namely

$$P_r(z) = z^{\beta_{rr}}(1+z^{\beta_{r-1,r}})\cdots(1+z^{\beta_{1r}}),$$

the expansion of which yields 2^{r-1} terms whose exponents define the 2^{r-1} positive transitions of g_r. In this way, an r-element loop-free network may be completely characterized in terms of the sequence of polynomials $P_1(z)$, $P_2(z), \cdots, P_r(z)$, provided only that the sequence of ordered exponents in the factored form of each $P_k(z)$ corresponds to the sequence of input weights on element k in accordance with the numbering of the elements in the network. This condition will be met automatically if

$$\beta_{jk} \leq \beta_{j-1,k}$$

for $j=1, 2, \cdots, k$, $k=1, 2, \cdots, r$. Further, the maximum number of transitions will be achieved provided only that the sequence of positive transition points alternate with the sequence of negative transition points for each $g_k(z)$. That is, the ordered exponents of each $P_k(z)$ in expanded form must alternate with the ordered exponents of the set of *all* previous polynomials, $P_1(z), P_2(z), \cdots, P_{k-1}(z)$.

An example in which this alternation condition is satisfied is provided by the network of three elements in which

$$\beta_{11}=5 \qquad \beta_{12}=4 \qquad \beta_{13}=5.$$
$$\beta_{22}=3 \qquad \beta_{23}=3$$
$$\beta_{33}=1.$$

For this case

$$P_1(z) = z^5$$
$$P_2(z) = z^3 + z^7$$
$$P_3(z) = z + z^4 + z^6 + z^9.$$

Thus the exponents (3, 7) of P_2 alternate with the exponent (5) of P_1, and the exponents (1, 4, 6, 9) of P_3 alternate with the exponents (3, 5, 7) of P_1 and P_2. Hence these three polynomials define a g_3 function with

[6] R. A. Short, "A Theory of Relations Between Sequential and Combinational Realizations of Switching Functions," Stanford Electronics Lab., Stanford University, Stanford, Calif., Tech. Rept. No. 098–1; December, 1960.

transition at $x=1, 3, 4, 5, 6, 7$, and 9. If $n=10$, for example, then

$$g_3 = S_{1,2,4,6,9,10}(x_1, \cdots, x_{10}).$$

The parity function $f_p(x_1, \cdots, x_n)$ introduced above provides another example in which the alternation condition is satisfied. From the values $\beta_{jk}=2^{r-j}$ $(j \leq k,\ j, k=1, 2, \cdots, r)$, we have

$$P_k(z) = z^{2^{r-k}}(1 + z^{2^{r-k+1}}) \cdots (1 + z^{2^{r-2}})(1 + z^{2^{r-1}}), \qquad k = 1, 2, \cdots, r,$$

so that the exponents of

$$P_r(z) = z + z^3 + z^5 + \cdots + z^{2^r-1}$$

locate the positive transitions in g_r, and the exponents of all lower-order $P_k(z)$, namely,

$$\sum_{k=1}^{r-1} P_k(z) = z^2 + z^4 + z^6 + \cdots + z^{2^r-2}$$

locate the negative transitions in g_r.

Consider now the consequences of a violation of the alternation condition on the exponents of successive $P_k(z)$. With reference to Fig. 4(b), if threshold β_{22} were increased so that

$$\beta_{11} \geq \beta_{22} \geq \beta_{33} + \beta_{23},$$

the second and third transition points would pass and cancel, merging the first two "pulses" of g_3 into a single pulse. On the other hand, if β_{22} were reduced so that

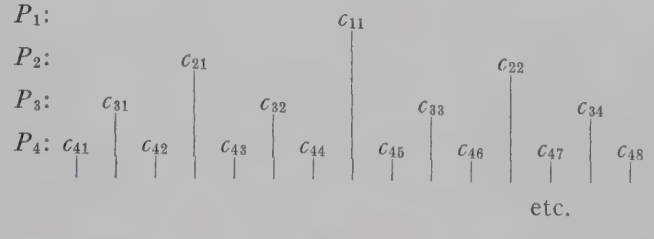
$$\beta_{22} \leq \beta_{33},$$

the first and second transition points would pass and cancel, eliminating the first pulse entirely. In general, if a pair of positive and negative transition points in g_k ever occur out of order, they will cancel out, eliminating either a pulse or an interpulse gap, and modifying the pattern of transitions in successive element outputs $g_{k+1}, \cdots, g_r$.

The succession of g functions corresponding to a sequence of polynomials $P_1(z), P_2(z), \cdots, P_r(z)$ for which the alternation condition is not necessarily satisfied may be derived by the following procedure. Let us place in an r-row array the exponents c_{kj} of the expanded polynomials,

$$P_k(z) = z^{ck1} + z^{ck2} + \cdots + z^{ck2^k-1}$$

keeping the terms in the same order in which they naturally occur as a result of the expansion of the product (not necessarily in order of increasing value):



Note that row k of this array may be thought of as being divided into 2^k-intervals by the set of exponents in row k and the rows above it. We now enter in these intervals the values assumed by the g functions, successively in rows $1, 2, \cdots, r$ for $g_1, g_2, \cdots, g_r$. Enter a *0* to the left and a *1* to right of c_{11}. In row k, we note for each interval whether or not the exponents defining the ends of the interval increase from left to right. (We imagine a *0* and an ∞ at the extreme left and right ends of the array.)

1. For odd-numbered intervals, enter a *0* if they increase, and a *1* otherwise.
2) For even-numbered intervals, enter a *1* if they increase, and a *0* otherwise.
3) In every interval in which an increase did not occur, delete all exponents c_{ij} in this same interval which fall in rows *below* row k.
4) Proceed to row $(k+1)$.

The values of the exponents in the bottom row $[P_r(z)]$ which fall between a *0* and a *1* are now the positive-transition x values in g_r, and those (leftmost) exponents in other rows which fall between a *1* and a *0* in the bottom row are the negative-transition x values in g_r.

The validity of this procedure follows from the recognition of c_{ki} as the apparent threshold of element k when the previous g outputs $(g_1, g_2, \cdots, g_{k-1})$ are regarded as the binary form of the number i $(i=1, 2, \cdots, 2^{k-1})$. Consequently, a positive transition can occur at $x=c_{ki}$ if and only if its value falls between the limits of the interval of the previous row in which it is positioned in the array. The deleted terms arise when and only when a particular sequence of g outputs does not occur.

The network of Fig. 5, for example, can be completely described in terms of the polynomials

$$P_1(z) = z^{13}$$

$$P_2(z) = z^6(1 + z^{20}) = z^6 + z^{26}$$

$$P_3(z) = z^4(1 + z)(1 + z^{12}) = z^4 + z^5 + z^{16} + z^{17}$$

$$P_4(z) = z^2(1 + z^3)(1 + z^8)(1 + z^{16})$$

$$= z^2 + z^5 + z^{10} + z^{13} + z^{18} + z^{21} + z^{26} + z^{29}.$$

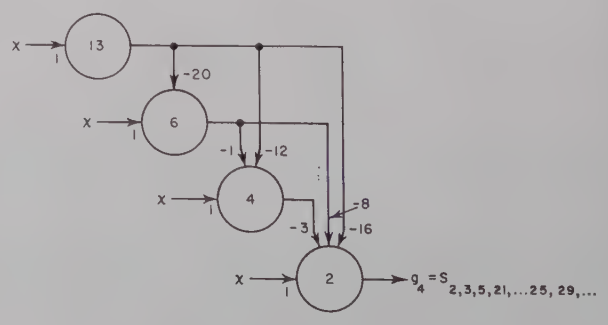
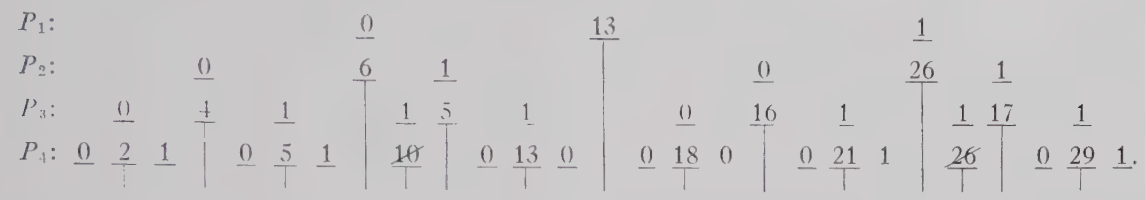


Fig. 5—An example of a network in which the alternation condition is not satisfied.

The array of exponents, with g values entered in accordance with the steps above, then appears as follows:



Thus, g_4 has positive transitions at $x=2$, 5, 21, and 29, and negative transitions at $x=4$, 6, and 26. Note the two deletions in row 4 which occurred because of the decreases over the 3rd and 7th intervals in row 3.

In this way, the output $f=g_r$ of any feed-forward network of linear-input elements may be determined with a minimum of calculation. It is apparent that, while a succession of misorderings of the exponent values will always reduce the number of pulses in g_r, it will allow at the same time fewer constraints between the remaining transition x values because of the flexibility in the amount of overlap of pulses and interpulse gaps, and freedom of choice of which pulses are eliminated. We will now see how this flexibility may be used to advantage.

Some Remarks on Synthesis

The synthesis problem is one of deriving from a given symmetric function at least one network (that is, a set of β values) which realizes this function over the range $0\leq x\leq n$ with a minimum number $r_{\min}$ of elements. While the synthesis process for arbitrary n appears to be very intricate, all symmetric functions of up to about twelve variables can be handled fairly easily by inversion of the above analysis procedure. In fact, symmetric functions of *any* number of variables, and having a total number τ of transitions up to about nine, can be synthesized in this way.

We have already seen that an r-element network can give rise to no more than 2^{r-1} positive transitions, or 2^r-1 transitions in all, in the output function g_r. Therefore, for all symmetric functions

$$r_{\min}(\tau) \geq 1 + [\log_2(\tau)].$$

This lower bound is achieved for the parity function, for which $\tau=n$, so the bound cannot be improved.

Minnick[1] describes a procedure for the synthesis of an arbitrary symmetric function which provides the upper bound

$$r_{\min}(\tau) \leq 1 + [\tau/2].$$

A simple argument shows that this bound cannot be improved. In an r-element network, only r of the 2^{r-1} possible positive transitions in g_r are arbitrary, since the entire set is determined by the r constants $\beta_{1r}, \beta_{2r}, \cdots, \beta_{rr}$. Thus by choosing successive positive-transition values large enough, we can certainly find a symmetric function having just r positive transitions, in which the other $2^{r-1}-r$ potential positive transitions are not desired and must be cancelled by negative transitions produced by the other $r-1$ elements. If the largest transition is positive (τ odd), then $\tau\leq 2r-1$. If the largest transition is negative (τ even), there will be a hidden positive transition at some value of $x>n$ (since $g_r\to 1$ as $x\to\infty$), so $\tau\leq 2r-2$. In either case, $r\geq 1+[\tau/2]$ for this function, and the tightness of the bound is established.

Thus, the minimum number $r_{\min}$ of elements required for the realization of any symmetric function having τ transitions in value as x increases from 0 to n satisfies the inequality

$$1 + [\log_2(\tau)] \leq r_{\min}(\tau) \leq 1 + [\tau/2],$$

and this range cannot be further narrowed. Since $\tau\leq n$, and $\tau=n$ only for the parity function, the same inequality is valid if τ is replaced by n, except that the upper bound is no longer tight, and indeed may be rather poor:

$$1 - [\log_2(n)] \leq r_{\min}(n) < 1 + [n/2].$$

Values of these two bounds are listed in the upper rows of Table I. Note that the bounds are equal up to $\tau=5$.

Analysis of numerous special cases leads to the results summarized in Table I, which lists for each pair of values of n and τ the number $r_{\min}$ of elements which is required in the minimal realization. The number in parentheses following an entry $r_{\min}$ indicates the number of different symmetric functions (symmetry types) whose minimal realization requires just $r_{\min}$ elements. When no such number is given, all $\binom{n}{\tau}$ symmetric functions of n variables having τ transitions require the same number $r_{\min}$ of elements.[7]

Let the transition x values of a given function to be synthesized be (in increasing order) $a_1, b_1, a_2, b_2, a_3, \cdots$. At best, no cancellations are necessary, and the polynomial

$$z^{a_1} + z^{a_2} + z^{a_3} + z^{a_4} + z^{a_5} + \cdots$$

[7] An argument which we do not present here reveals that the number of 6-transition, 3-element symmetry types for any value of n greater than 6 exceeds the corresponding number for $n-1$ by the sum $\Sigma\alpha_1\alpha_2\alpha_3$ over all additive partitions of $n-3$ into just three positive integers, the first of which is at least as big as the second: $n-3=\alpha_1+\alpha_2+\alpha_3$, $\alpha_1\geq\alpha_2>0$, $\alpha_3>0$. A related expression can be derived when $\tau=7$.

TABLE I

MINIMAL NUMBER r_{min} OF LINEAR-INPUT ELEMENTS REQUIRED FOR THE REALIZATION OF SYMMETRIC SWITCHING FUNCTIONS OF n VARIABLES HAVING τ TRANSITIONS IN VALUE

	τ:	1	2	3	4	5	6	7	8	9	10	11
	Lower bound	1	2	2	3	3	3	3	4	4	4	4
	Upper bound	1	2	2	3	3	4	4	5	5	6	6
n	1	1										
	2	1	2									
	3	1	2	2								
	4	1	2	2	3							
	5	1	2	2	3	3						
	6	1	2	2	3	3	3					
	7	1	2	2	3	3	3 (5) 4 (2)	3				
	8	1	2	2	3	3	3 (19) 4 (9)	3 (4) 4 (4)	4			
	9	1	2	2	3	3	3 (53) 4 (31)	3 (14) 4 (22)	4	4		
	10	1	2	2	3	3	3 (129) 4 (81)	3 (36) 4 (84)	4	4	4	
	11	1	2	2	3	3	3 (275) 4 (187)	3 (84) 4 (246)	4	4 (47) 5 (8)	4	4

can be put into factorable form as $P_r(z)$ merely by adding as necessary terms with exponents x outside of the range $0 \leq x \leq n$. The degrees of the factors may then be identified with the weights β_{jr}, as defined previously. For example, the symmetric function

$$f_s = S_{1,2,4,6,7,10,12}(x_1, \cdots, x_{13})$$

has transition values

$$a_1 = 1 \quad a_2 = 4 \quad a_3 = 6 \quad a_4 = 10 \quad a_5 = 12$$
$$b_1 = 3 \quad b_2 = 5 \quad b_3 = 8 \quad b_4 = 11 \quad b_5 = 13,$$

and $P_4(z)$ may be formed as follows, with the original terms in parentheses:

$$P_4(z) = z^{-1} + (z + z^4 + z^6 + z^{10} + z^{12}) + z^{15} + z^{17}$$
$$= z^{-1}(1 + z^2)(1 + z^5)(1 + z^{11}),$$

so

$$\beta_{44} = -1, \quad \beta_{34} = 2, \quad \beta_{24} = 5, \quad \text{and} \quad \beta_{14} = 11.$$

$P_3(z)$ may then be formed from alternate terms of the sequence of negative transitions: 0, 3, 5, 8, 11, 13, $\cdots$. Thus

$$P_3(z) = z^0 + (z^5 + z^{11}) + z^{16} = z^0(1 + z^5)(1 + z^{11}),$$

so

$$\beta_{33} = 0, \quad \beta_{23} = 5, \quad \text{and} \quad \beta_{13} = 11.$$

Similarly,

$$P_2(z) = z^3 + z^{13} = z^3(1 + z^{10})$$
$$P_1(z) = z^8,$$

so $\beta_{22} = 3$, $\beta_{12} = 10$, $\beta_{11} = 8$. The network with these β values then realizes the given symmetric function.

In general, the possibility of cancellation of terms must be taken into account, in which case the requirement of alternating terms in successive $P_k(z)$ must be replaced by a more detailed analysis of the network under consideration.

It can be seen from Fig. 4(b) and the form of $P_3(z)$ that if $r \leq 3$, all of the transition x values of g_1, g_2, and g_3 are completely arbitrary except one of the positive transitions (*e.g.*, the last one) of g_3. Thus, all symmetric functions of five or fewer transitions can be realized with the minimum number of elements. From the form of the polynomial $P_3(z)$, the four positive transition x values must satisfy the equality $a_4 - a_3 = a_2 - a_1$. For $\tau = 6$, therefore, only those functions for which n falls between the third and fourth pulses are realizable with three elements:

$$b_3 \leq n < a_3 + a_2 - a_1,$$

and all others require four elements. For $\tau = 7$, the equality

$$a_4 = a_3 + a_2 - a_1$$

must hold if three elements are to be adequate, otherwise, four elements are needed.

For $\tau > 7$, at least four elements are necessary. The selection of a set of β values to produce the desired sequence of transitions, with cancellations as necessary, and simultaneously to satisfy the multiplicative condition implicit in the polynomial expansion, may be exe-

cuted with a trial-and-error process. As an example, consider the function

$$f_5 = S_{1,3,5,8}(x_1, \cdots, x_{11}),$$

which has the transition values

$$a_1 = 1 \quad a_2 = 3 \quad a_3 = 5 \quad a_4 = 8$$
$$b_1 = 2 \quad b_2 = 4 \quad b_3 = 6 \quad b_4 = 9.$$

Since $\tau = 8$, the bounds indicate that either 4 or 5 elements are required. With $r = 4$, the four a_k might be identified with the transition points β_{44}, $\beta_{44}+\beta_{34}$, $\beta_{44}+\beta_{24}$, and $\beta_{44}+\beta_{14}$, giving

$$P_4(z) = z(1 + z^2)(1 + z^4)(1 + z^7)$$
$$= z + z^3 + z^5 + z^7 + z^8 + z^{10} + z^{12} + z^{14},$$

but then unwanted positive transitions occur at $x = 7$, 10, 12, and 14. For $n = 11$, the first two must be cancelled with negative transitions, but this can be done with the set of β values expressed in the polynomials

$$P_3(z) = z^2 + z^6 + z^9 + z^{13} = z^2(1 + z^4)(1 + z^7)$$
$$P_2(z) = z^4 + z^{10} = z^4(1 + z^6)$$
$$P_1(z) = z^7.$$

The final network is shown in Fig. 6(a), and has four elements.

If the number n of variables had been greater than 11, other β values might be selected to cancel the positive transitions at 12 and 14, or to yield the desired transitions in another way. Trial of all of these possibilities, which are not very numerous, reveals that no such selection is possible with only four elements, however, so that five elements are required for this case. One possible network, valid for $n \geq 9$ and minimal for $n \geq 12$, is shown in Fig. 6(b).

Any symmetric function for which $\tau \leq 9$, and many others, can be handled easily by this approach. The cases $\tau = n$, $\tau = n-1$, and $\tau = n-2$ have either been considered already or are not difficult. Thus all cases for which $n \leq 12$ are amenable to the procedure described with a very modest amount of trial and error effort.

The previously mentioned procedure due to Minnick[1] yields a network whose $r = 1 + [\tau/2]$ elements are arranged as shown in Fig. 7(a), with the following weights and thresholds:

$$\beta_{jk} = 0 \qquad j < k < r$$
$$\beta_{jr} = a_{j+1} - a_1 \qquad j < r$$
$$\beta_{kk} = b_k \qquad k < r.$$
$$\beta_{rr} = a_1$$

Thus all elements except the rth are driven only by x inputs and drive only the rth element, which provides the output. An alternative configuration is shown in Fig. 7(b). Several circuit arrangements which are hybrid combinations of the general feed-forward structure and one of these circuits of Fig. 7 are possible, and can

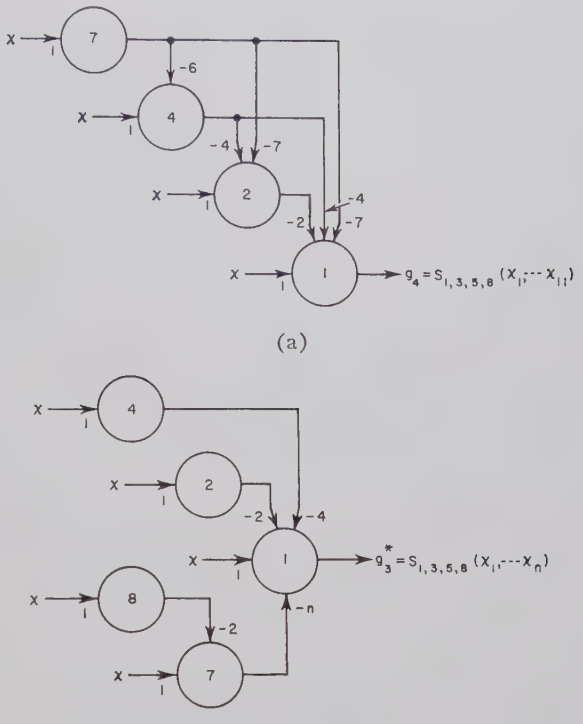


Fig. 6—Minimal linear-input networks for realization of the example $S_{1,3,5,8}(x_1, \cdots, x_n)$ for (a) $n < 12$, and (b) $n \geq 12$.

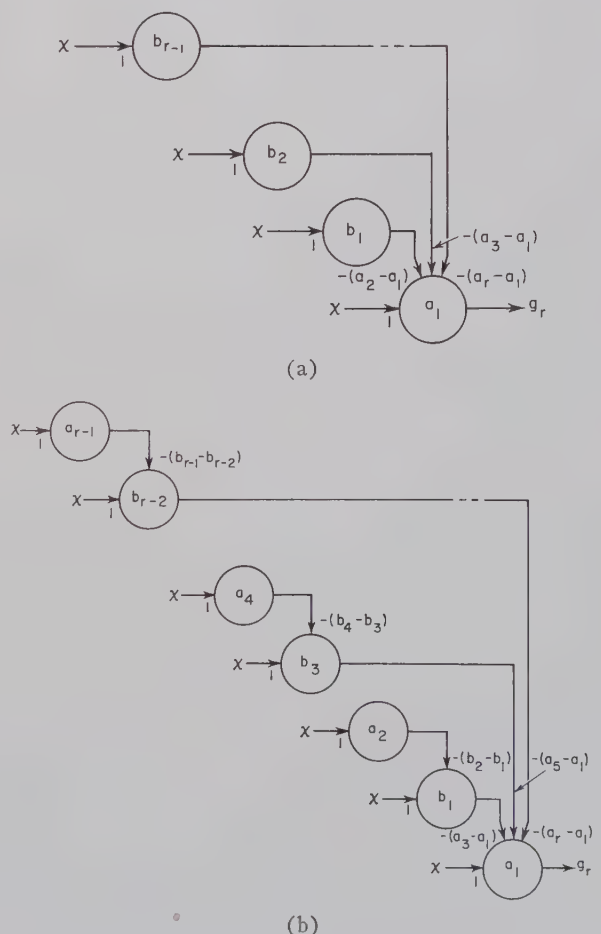


Fig. 7—Two alternative realizations which use a number of elements equal to the upper bound.

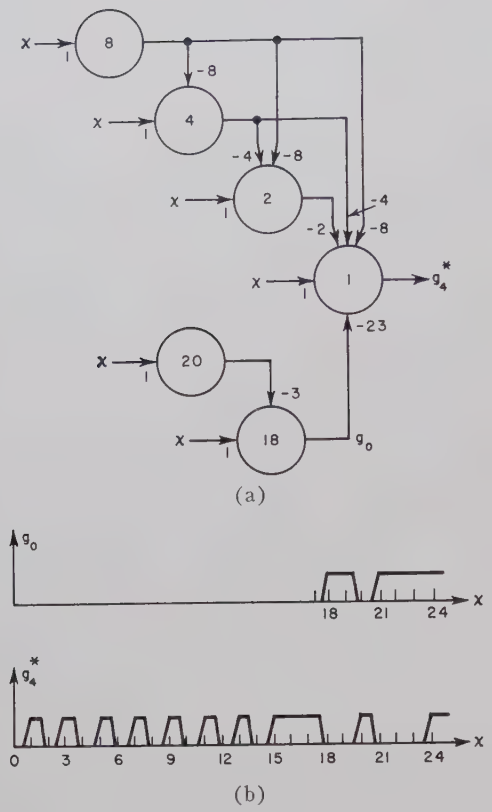


Fig. 8—An example of a hybrid realization and the element-output functions of its two main parts.

be applied in cases when τ is large to yield a network which is not necessarily minimal, but may represent a saving over the number of elements required for the upper bound. One example of a hybrid structure is provided by the example of Fig. 6(b), introduced earlier. The network of Fig. 8 provides another example: here $\tau = 19$, so that $r_{\min} \leq 10$. The fact that the polynomial condition is not satisfied for the entire set of positive transitions, but is satisfied for the first eight, suggests that the last few pulses in g_r be realized separately. The hybrid network requires $r = 6$ elements.

Discussion

While the approach to symmetric-function synthesis presented in the previous section provides solutions to all cases of any conceivable practical importance, there nevertheless remains the challenging theoretical problem of developing a general procedure for the minimal realization of arbitrary symmetric functions. As is frequently typical of synthesis algorithms, the main motivation for their development probably lies more in the understanding to be gained of the structure and properties of linear-input networks than in the possible direct utility of the method itself. In the present case, the knowledge of just how economical networks of linear-input elements can give rise to a desired type of symmetric-terminal behavior could very well provide much of the basis and insight needed for a solution of the synthesis problem for arbitrary switching functions.

The reader may have already observed the close relation between the method described and Markov's procedure for the synthesis of minimum-NOT networks.[8] A direct element-for-element conversion of Markov networks to linear-input networks, even for the symmetric-function case, generally does not appear to be possible, and, when it is possible, it does not necessarily lead to minimal linear-input networks. Nevertheless, the method of decomposition of the symmetric function into a succession of unate subfunctions which can be optimally combined is common to both procedures.[3] The possible extension of the above procedure to arbitrary switching functions through a development analogous to Markov's procedure for arbitrary functions suggests that the given function be decomposed into a succession of linear-input, rather than unate, subfunctions. (The two classes are identical when the given function is symmetric.) In this way, examples of non-symmetric switching functions of arbitrary complexity can be created which have a minimal realization for virtually any number of elements. These possibilities are presently under investigation.

It was pointed out to the author by D. A. Huffman of M.I.T. that an arbitrary switching function $f = \sum(a_1, a_2, \cdots, a_s)$ of m variables $x_1, x_2, \cdots, x_m$ (expressed here in "decimal" form, following Caldwell[9]) can be written as a symmetric function in $2^m - 1$ variables, $f = S_{a_1,a_2,\cdots,a_s}(x_1, x_2, x_2, x_3, x_3, x_3, x_3, x_4, \cdots, x_m.)$ Here variable x_i occurs with multiplicity 2^{i-1}. Thus, this equivalence enables the procedure of this paper to be applied to the synthesis of arbitrary switching functions, although, of course, the final result will in general not be very economical in the number of elements required.

Finally, it should be noted that for the networks under consideration, most or all of the element non-x input weights are *negative*. Certain circuit realizations of the linear-input element—*e.g.*, those using resitive adders with transistors—allow negative weights to be realized much more easily than positive weights, and therefore might be particularly desirable from an applied point of view. Since the x inputs all have positive weights and are common to all elements, these inputs can be collected and summed separately, then applied through an inverting amplifier to the entire set of elements.

Acknowledgment

The author is indebted to R. Minnick of Stanford Research Institute, whose work stimulated his serious interest in the subject area of this paper, and with whom many profitable technical discussions have taken place.

[8] A. A. Markov, "On the inversion complexity of a system of functions," *J. Assoc. Computing Mach.*, vol. 5, pp. 331–334, October, 1958; translated from *Doklady Akad. Nauk S.S.S.R.*, vol. 116, pp. 917–919, 1957.

[9] S. H. Caldwell, "Switching Circuits and Logical Design," John Wiley and Sons, Inc., New York, N. Y.; 1958. See especially p. 124.

Orthogonal Functions for the Logical Design of Switching Circuits*

ROBERT P. COLEMAN†, MEMBER, IRE

Summary—A new approach to the mathematical representation of switching functions is presented. It was developed in connection with a theoretical study of magnetic-core logic, but the results are considered to be more basic and general than the core-logic problem. The ampere-turns (MMF) expression for core switching is shown to be part of a special type of Fourier series expansion of a switching function, in which the turns are directly related to the spectrum of the function. Fouriers transform methods, used for analysis of *X*-ray diffraction, have been adapted to the representation of switching functions. The method leads not to Boolean algebra, but to ordinary algebra in terms of the orthogonal functions

$$(-1)^{k_1x_1+k_2x_2+\cdots+k_nx_n},$$

where

$$x_1, x_2, \cdots, x_n = 0, 1,$$

and

$$k_1, k_2, \cdots, k_n = 0, 1.$$

Methods of application are described for magnetic-core logic and for character recognition.

Introduction

A new approach to the mathematical representation of switching functions is presented. It was developed in connection with a theoretical study of magnetic-core logic, but the results are considered to be more basic and general than the core-logic problem. A problem of some importance is to find a simple test for the realizability of a given logic function with one magnetic core. This problem has not been completely solved, but consideration of it has led to some useful conceptions.

Core Switching

The switching of a magnetic core is determined by the value of MMF expressed in the form

$$F - F_0 = N_0x_0 + N_1x_1 + N_2x_2 + \cdots N_nx_n,$$

where

$x_1, x_2, \cdots, x_n$ denote input currents
$N_1, N_2, \cdots, N_n$ denote number of turns
N_0x_0 refers to a constant bias MMF.

The core will switch if F is greater than some threshold value F_0. A logic function can be realized with one core if it is possible to choose a set of values of N_0, N_1, N_2, $\cdots$, N_n (positive or negative) such that $F-F_0>0$ for all sets of values of inputs $x_1, x_2, \cdots, x_n$ for which the function denotes "truth," and $F-F_0<0$ for all other sets of values of the input variables.

This result has a useful geometrical interpretation. A well-known representation of a logic function is the assignment of truth values 1 or 0 to the vertices of a unit hypercube in n-dimensional space, having coordinate axes $x_1, x_2, \cdots, x_n$.

The equation

$$N_0x_0 + N_1x_1 + N_2x_2 + \cdots + N_nx_n = 0$$

is the equation of a hyperplane in this n-dimensional space. The condition for realizability of a function with one core is therefore equivalent to the condition that it be possible to pass a hyperplane through the hypercube in some position such that all vertices having the truth-value 1 are on one side of the hyperplane and all vertices having the truth-value 0 are on the other side of the hyperplane. Under these conditions the numbers N_1, N_2, $\cdots$, N_n are proportional to the components of a vector that is normal to the hyperplane.

X-Ray Diffraction

This picture suggests a similarity to problems of X-ray diffraction in which a crystal lattice composed of different kinds of atoms produces X-ray diffraction patterns that depend upon the orientation and spacing of various planes in the crystal lattice. The vertices of a unit hypercube can be considered as points of a space lattice, and the unit may be repeated periodically along all coordinate directions to produce an infinite periodic lattice. Values 0 or 1 associated with the lattice points define a periodic function of period 2 in each coordinate direction.

The relevant formulas of X-ray diffraction analysis are summarized in the Appendix. The application of this type of analysis to switching functions leads to the concept of the structure factor which may be called the spectrum of a switching function. The structure factor or spectral function contains all the information that is contained in the truth table of a switching function. This representation suggests a variety of conceptions in logical design. For example, in magnetic-core logic, the spectral function is directly related to the number of turns of the various windings on a core.

More generally, a switching function may be mechanized with other components in a circuit in which design parameters are directly related to the spectral function.

* Received by the PGEC, June 24, 1960; revised manuscript received, April 10, 1961.
† Research Center, Burroughs Corp., Paoli, Pa.

Orthogonal Functions

It is common to represent a switching function of n variables as an assignment of values 1 or 0 to the vertices of an n-dimensional hypercube. In the present development this representation is extended by considering a periodic function of an infinite space-lattice in n dimensions. The period is a distance of two along each coordinate axis. A switching function $f(\mathbf{x})$ can then be expanded by treating the values 1 or 0 as "atomic scattering factors" in the sense of X-ray diffraction analysis (Appendix). Since a period of two in the components of $\mathbf{x}$ corresponds to a period of one in the components of $\boldsymbol{R}$, the "structure-factor" $F(\boldsymbol{k})$ of a switching function is obtained by a simple modification of (3) of the Appendix as

$$F(\boldsymbol{k}) = \sum_{\mathbf{x}} f(\mathbf{x}) \exp\left(\frac{2\pi i \mathbf{x}\cdot\boldsymbol{k}}{2}\right).$$

The relation

$$\exp(\pi i) = -1$$

reduces this equation to the form

$$F(\boldsymbol{k}) = \sum_{\mathbf{x}} f(\mathbf{x})(-1)^{\boldsymbol{k}\cdot\mathbf{x}}.$$

Similarly, the fact that

$$\exp(-\pi i) = -1$$

and that the volume of the "unit-cell" is $V=2^n$, for n dimensions, leads to the inverse transform

$$f(\mathbf{x}) = \frac{1}{2^n}\sum_{\boldsymbol{k}} F(\boldsymbol{k})(-1)^{\boldsymbol{k}\cdot\mathbf{x}},$$

where the components of $\boldsymbol{k}$ take the values 0 and 1. These values of $\boldsymbol{k}$ are sufficient to determine $f(\mathbf{x})$ at the lattice points.

In later sections of this discussion, the representation of a switching function $f(\mathbf{x})$ as an expansion in terms of the orthogonal functions $(-1)^{\boldsymbol{k}\cdot\mathbf{x}}$ will be made the basis of various logical design problems.

The functions

$$(-1)^{\boldsymbol{k}\cdot\mathbf{x}}$$

are called orthogonal because for $\boldsymbol{k}' \neq \boldsymbol{k}$

$$\sum_{\mathbf{x}} (-1)^{\boldsymbol{k}\cdot\mathbf{x}}(-1)^{\boldsymbol{k}'\cdot\mathbf{x}} = 0.$$

If $\boldsymbol{k}' = \boldsymbol{k}$, this expression has the value

$$\sum_{\mathbf{x}} (-1)^{2\boldsymbol{k}\cdot\mathbf{x}} = 2^n,$$

for n variables, $x_1, x_2, \cdots, x_n$.

Matrix Form of Transformation

The transformation equations

$$F(\boldsymbol{k}) = \sum_{\mathbf{x}} f(\mathbf{x})(-1)^{\boldsymbol{k}\cdot\mathbf{x}}$$

$$f(\mathbf{x}) = \frac{1}{2^n}\sum_{\boldsymbol{k}} F(\boldsymbol{k})(-1)^{\boldsymbol{k}\cdot\mathbf{x}}$$

can conveniently be written in matrix form. For example, for $n=2$,

$$\begin{vmatrix} F(0,0) \\ F(0,1) \\ F(1,0) \\ F(1,1) \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{vmatrix} \begin{vmatrix} f(0,0) \\ f(0,1) \\ f(1,0) \\ f(1,1) \end{vmatrix}$$

$$\begin{vmatrix} f(0,0) \\ f(0,1) \\ f(1,0) \\ f(1,1) \end{vmatrix} = \frac{1}{4} \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{vmatrix} \begin{vmatrix} F(0,0) \\ F(0,1) \\ F(1,0) \\ F(1,1) \end{vmatrix}.$$

By matrix multiplication, it may be verified that

$$\frac{1}{4}\begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{vmatrix} \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

This example confirms the fact that one transformation is the inverse of the other one.

Magnetic Core Logic

So far we have derived a representation for a switching function involving operations of ordinary multiplication and addition, rather than Boolean operations. The linear-input expression (MMF) for a magnetic core is also a function involving operations of ordinary multiplication and addition.

If a given switching function is realizable with a single core, we know that values of N_i exist, such that the hyperplane,

$$\sum_{i=0}^{n} N_i x_i = 0,$$

partitions the ones from the zeros of the function. As long as the x_i take only the values 0 and 1, the substitution

$$1 - 2x_i = (-1)^{x_i}$$

can be used to express the hyperplane equation in the form

$$a_0 + \sum_{i=1}^{n} a_i(-1)^{x_i} = 0,$$

where

$$N_0 x_0 = a_0 + \sum_{i=1}^{n} a_i$$

$$N_i = -2a_i, \qquad i \neq 0.$$

We have thus expressed the input of a core in the form of a sum of linear orthogonal functions

$$S(\mathbf{x}) = a_0 + \sum_{i=1}^{n} a_i(-1)^{x_i},$$

and we have expressed the output of a core in a similar form, but containing both linear and nonlinear terms,

$$f(\mathbf{x}) = \frac{1}{2^n} \sum_{k} F(\mathbf{k})(-1)^{\mathbf{k}\cdot\mathbf{x}},$$

where

$$F(\mathbf{k}) = \sum_{x} f(\mathbf{x})(-1)^{\mathbf{k}\cdot\mathbf{x}}.$$

We now introduce the idea of thinking of the input $S(\mathbf{x})$ as an approximation to the function $f(\mathbf{x})$. Only those functions are realizable with a single core for which there exists an input function $S(\mathbf{x})$ that is a *suitable* linear approximation to $f(\mathbf{x})$. The difficulty of the problem has now been concealed in the word "suitable." Although we do not have an explicit formula for a "suitable" linear approximation, we have found some interesting properties of the simplest approximation, namely, the approximation formed by equating $S(\mathbf{x})$ to the linear part of $f(\mathbf{x})$. It is easily shown that this procedure furnishes a best approximation in a least squares-error sense. The value of the residual squared error is easily evaluated and it furnishes an indication (but not a rigorous test) for realizability of a switching function with one core.

Least-Squares Error Property

Define the squared error by the expression

$$E = \sum_{x} [f(\mathbf{x}) - S(\mathbf{x})]^2,$$

in which the summation is carried out over the whole truth table of the given function $f(\mathbf{x})$. From the orthogonal property of the functions $(-1)^{x_i}$, it follows that

$$\sum_{x} S^2(\mathbf{x}) = 2^n \left[a_0^2 + \sum_{i=1}^{n} a_i^2 \right].$$

Hence,

$$E = \sum_{x} f^2(\mathbf{x}) - 2 \sum_{x} f(\mathbf{x}) \left[a_0 + \sum_{i} a_i(-1)^{x_i} \right] + 2^n \left[a_0^2 + \sum_{i=1}^{n} a_i^2 \right].$$

The conditions for minimum E are

$$0 = \frac{\partial E}{\partial a_0} = -2 \sum_{x} f(\mathbf{x}) + 2a_0 2^n$$

$$0 = \frac{\partial E}{\partial a_i} = -2 \sum_{x} f(\mathbf{x})(-1)^{x_i} + 2a_i 2^n,$$

or

$$a_0 = \frac{1}{2^n} \sum_{x} f(\mathbf{x})$$

$$a_i = \frac{1}{2^n} \sum_{x} f(\mathbf{x})(-1)^{x_i}.$$

Hence, the least-squares procedure gives the same values of coefficients as the inversion formula of the orthogonal expansion.

Value of Residual Squared Error

If the expression for the coefficients is combined with the expression for E, the following expression for minimum-squared error is obtained:

$$E_{\min} = \sum_{x} f^2(\mathbf{x}) - 2^n \left[a_0^2 + \sum_{i=1}^{n} a_i^2 \right],$$

where

$$a_0 = \frac{1}{2^n} \sum_{x} f(\mathbf{x})$$

$$a_i = \frac{1}{2^n} \sum_{x} f(x)(-1)^{x_i}.$$

A more homogeneous expression is obtained if we deal with the function

$$\phi(\mathbf{x}) = 1 - 2f(\mathbf{x}),$$

in which it is clear that $\phi(\mathbf{x})$ takes values $+1$ and -1 corresponding to values 0 and 1 for $f(\mathbf{x})$. Then the error expression for $\phi(\mathbf{x})$ is

$$E'_{\min} = \sum_{x} \phi^2(\mathbf{x}) - 2^n \left[b_0^2 + \sum_{i=1}^{n} b_i^2 \right],$$

where

$$b_0 = \frac{1}{2^n} \sum_{x} \phi(\mathbf{x})$$

$$b_i = \frac{1}{2^n} \sum_{x} \phi(\mathbf{x})(-1)^{x_i};$$

but

$$\phi^2(\mathbf{x}) = 1,$$

$$\sum_{x} \phi^2(\mathbf{x}) = 2^n,$$

and the error expression can be written as

$$\frac{E'_{\min}}{2^n} = 1 - \sum_{i=0}^{n} b_i^2.$$

This expression has the desirable properties of invariance with respect to operations of complementation of any of the variables x_i, or of the function, or of permutation of any of the variables.

In many examples this expression has been found to have a smaller value for one-core-realizable functions than for nonrealizable functions. A tentative critical value for the error expression has been obtained by considering the functions of a very large number of variables such that the function is true whenever at least half of the variables are true. For this case, with even values of n,

$$\frac{E'_{\min}}{2^n} = 1 - \frac{\binom{n}{n/2}(n+1)}{2^{2n}},$$

where

$\binom{n}{n/2}$ is a binomial coefficient.

This expression approaches the limit

$$1 - \frac{2}{\pi}, \quad \text{as} \quad n \to \infty.$$

We repeat that the value of the above error expression compared to the tentative critical value of $1-2/\pi$ is not a rigorous test of realizability with one core.

Character Recognition

A quite different field of application of these orthogonal functions is the logical design of character-recognition systems. The analysis for switching functions is made applicable to character recognition by relating the pattern of ink forming the character to the pattern of ones in a Veitch diagram. For example, Fig. 1 is a Veitch diagram for a function of six variables. The function, expressed in Boolean notation as

$$f = x_3\bar{x}_4 + \bar{x}_3x_4 + x_5x_6,$$

plots into a pattern of ones that resembles the shape of character 0.

The expansion

$$f(\mathbf{x}) = \frac{1}{2^6}\sum_{k} F(k)(-1)^{k\cdot x},$$

where

$$F(\boldsymbol{k}) = \sum_{x} f(\mathbf{x})(-1)^{k\cdot x},$$

therefore may be considered as a representation of the character in terms of 64 "harmonics" $F(\boldsymbol{k})$ rather than 64 points. It is expected that this will be an efficient representation, because a few of the harmonics will usually contain most of the information required for recognition.

For example, the character 0 is quite well represented by the expression

$$f(\mathbf{x}) = \frac{1}{64}\left[40 - 24(-1)^{x_3+x_4} - 8(-1)^{x_5} - 8(-1)^{x_6}\right],$$

which approximates the above Boolean expression. Values of this approximate expression, tabulated in the squares of a Veitch diagram, are shown in Fig. 2.

It is also expected that the "noise" will predominate in the "high harmonics," and that the "signal" will predominate in the "low harmonics." Reliability of recognition therefore is expected to be improved by leaving out the "high harmonics."

It now seems to be feasible to make these principles more precise and explicit by the study of principles of optimum design in terms of the above concepts of the spectral function $F(\boldsymbol{k})$. A program of computation of values of $F(\boldsymbol{k})$ for statistical samples of "noisy characters" is also needed.

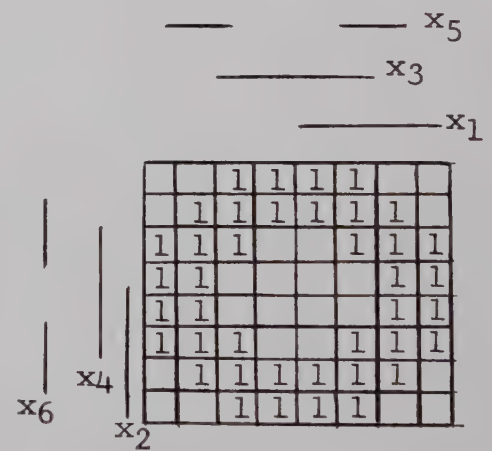
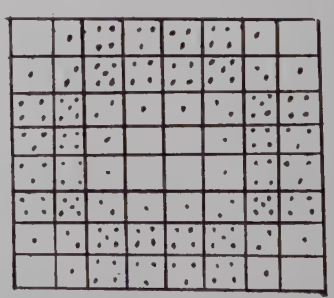


Fig. 1—The function $x_3\bar{x}_4+\bar{x}_3x_4+x_5x_6$ resembles the character 0 when represented on a Veitch diagram.

0	1	4	3	3	4	1	0
1	2	5	4	4	5	2	1
4	5	2	1	1	2	5	4
3	4	1	0	0	1	4	3
3	4	1	0	0	1	4	3
4	5	2	1	1	2	5	4
1	2	5	4	4	5	2	1
0	1	4	3	3	4	1	0

(a)



(b)

Fig. 2—Values of $4f(x)$ corresponding to $f(x)=\frac{1}{64}[40-24(-1)^{x_3+x_4}-8(-1)^{x_5}-8(-1)^{x_6}]$ tabulated as (a): Numbers. (b) Density of dots.

Appendix

Fourier Transforms in X-Ray Diffraction

In a crystal lattice the electron density $\rho(\boldsymbol{R})$ is a periodic function in the components of the position vector $\boldsymbol{R}$. The electron density $\rho(\boldsymbol{R})$ therefore can be represented by a triple Fourier series

$$\rho(R) = \frac{1}{V} \sum \sum_{-\infty}^{\infty} \sum F(\boldsymbol{k}) \exp(-2\pi i \boldsymbol{k} \cdot \boldsymbol{R}), \qquad (1)$$

where

V = volume of unit cell of crystal,
$\boldsymbol{k}$ = a vector, the components of which are integers,
$F(\boldsymbol{k})$ = the three-dimensional Fourier transform of $\rho(\boldsymbol{R})$; in X-ray literature the values of $F(\boldsymbol{k})$ for different values of $\boldsymbol{k}$ are called "structure factors."

The structure-factor equation is

$$F(\boldsymbol{k}) = \iiint \rho(\boldsymbol{R}) \exp(2\pi i \boldsymbol{R} \cdot \boldsymbol{k}) dV. \qquad (2)$$

The structure factor $F(\boldsymbol{k})$ is expressed conveniently in terms of the transforms of each individual atom. The latter are called the "atomic scattering factors," and are denoted by a symbol f_j for the jth atom.

The great advantage of this treatment lies in the fact that the integral over the point atom crystal can be replaced by a sum

$$F(\boldsymbol{k}) = \sum_{j}^{N} f_j \exp(2\pi i \boldsymbol{R}_j \cdot \boldsymbol{k}), \qquad (3)$$

where $\boldsymbol{R}_j$ are the coordinates of the jth atom ($j = 1, 2, \cdots, N$) and the summation is over the N atoms contained in the unit cell.

Autocorrelations for Boolean Functions of Noiselike Periodic Sequences*

B. M. EISENSTADT†, member, ire, and B. GOLD‡, senior member, ire

Summary—**One method of generating a waveform whose correlation function resembles that of noise is by means of combinations of periodic binary sequences. In this paper the properties of the correlation function for arbitrary functions of n periodic binary sequences are investigated. An especially simple formulation is obtained when each binary variable in all the sequences has equal probability of being 0 or 1. For this case, it is shown that there are only two functions which result in a correlation function like true purely random noise. One of these two functions corresponds to addition modulo 2. Also, the correlation for the case of a random function of n sequences is derived. Finally, expressions are obtained for the number of degenerate Boolean functions.**

I. Introduction

IN this paper we shall introduce a very general form of digit generator and investigate some of the properties of pseudo-random binary sequences produced by it. The digit generator is composed of n periodic sequences of binary digits, and the output is constructed by successively combining one digit at a time from each sequence, in accordance with a fixed rule.

Randomness is introduced by assuming that each sequence having some period P contains P 1's and 0's chosen independently in some random manner. The digit generator obtains its general form by allowing the fixed rule for constructing the output to be any Boolean function of the n binary digits from the constituent periodic sequences.

The results presented are an outgrowth of studies on noiselike properties of certain binary signals. Many of the results of the studies were contained in two reports[1,2] and dealt primarily with a very special but interesting rule for forming the output sequence, the addition modulo 2 of the digits from the periodic sequences. For this special case, Zierler[3] studied the mathematical properties of the output sequences, which he called linear recurring.

Three possible fields of application for this digit generator are communication, coding and computers. To our knowledge, the digit generator has been used in the first two fields.

There are many computer programming problems which involve the generation of pseudo-random se-

* Received by the PGEC, October 7, 1960; revised manuscript received, March 24, 1961.
† Appl. Res. Lab., Sylvania Electronics Systems, Waltham, Mass.
‡ Lincoln Lab., Mass. Inst. Tech., Lexington, Mass.

[1] B. Eisenstadt and B. Gold, "Correlation Functions of Certain Random Sequences," Lincoln Lab., Mass. Inst. Tech., Lexington, Group Rept. 34-25; July 28, 1954.
[2] L. Maximon, "Notes on Some Periodic Sequences of Random Variables," Lincoln Lab., Mass. Inst. Tech., Lexington, Group Rept. 34-23; March, 1954.
[3] N. Zierler, "Linear recurring sequences," *J. Soc. Ind. Appl. Math.*, vol. 7, pp. 32–48; March, 1959.

quences of digits. One direct application for this generator would be to produce sequences of digits for use in Monte Carlo methods, games theory, etc.

In this paper we concentrate on one particular property, the correlation function of the digit generator. A potentially important application of the theory developed in this paper is in testing systems or simulating system reponses to inputs of binary sequences where a varying correlation function is desired.

II. Formulation

First, we shall define the quantities and relationships necessary for our techniques. Next, we shall present systematic procedures illustrating our method of specifying the normalized mean-correlation function ρ_s, which is the vital characteristic under consideration for determining the corresponding subsets of the binary variables for a given truth table. A few special but interesting cases are investigated when the digits in the periodic sequences are 1 or 0 with equal probability. Finally, we shall examine the number of functions which always produce output sequences shorter than the maximum possible, and shall obtain an explicit result.

Let Y_k be a periodic sequence of binary random variables y_{jk}, where j goes from $-\infty$ to $+\infty$. The period of Y_k is p_k, so that $y_{j_1k}=y_{j_2k}$ if j_1-j_2 is a multiple of p_k. We stipulate that y_{j_1k} is independent of y_{j_2k} if j_1-j_2 is not a multiple of p_k and also that $y_{jk}=1$ with probability q and $y_{jk}=0$ with probability $1-q$ for all j and k.

Let X be a sequence of binary variables x_j such that each x_j is a function of $y_{j0}, y_{j1}, \cdots$, that is, of the jth y from each of the sequences $Y_0, Y_1, \cdots, Y_{n-1}$. Since each x_j is binary, it may be expressed most generally in the canonical form

$$x_j = \sum_{i=0}^{2^n-1} A_i H_{ji}. \tag{1}$$

H_{ji} is defined as

$$H_{ji} = \left[\prod_{k=0}^{n-1} (c_k y_{jk}) \right]_i. \tag{2}$$

$(c_k y_{jk})$ is interpreted as a single variable with the convention that $(c_k y_{jk})=y_{jk}$ if $c_k=1$ and $(c_k y_{jk})=y_{jk}'$ (the complement of y_{jk}) if $c_k=0$.

Each A_i belongs to a set Φ of 2^n binary variables. There are thus 2^{2^n} distinct combinations of all the A_i. Furthermore, i completely determines all the c_k for a given H_{ji}. In fact, the ordered set of c_k for a given i forms a binary number whose decimal equivalent is i. For example, if $n=4$ and $i=5$, then $c_0=0$, $c_1=1$, $c_2=0$, $c_3=1$, so that the binary number $c_0c_1c_2c_3=0101=5$. Thus, $H_{j5}=y_{j0}'y_{j1}y_{j2}'y_{j3}$.

For $n=3$ (1) is expanded as

$$\begin{aligned} x_j = {} & A_0 y_{j0}' y_{j1}^1 y_{j2}' + A_1 y_{j0}' y_{j1}' y_{j2} + A_2 y_{j0}' y_{j1} y_{j2}' \\ & + A_3 y_{j0}' y_{j1} y_{j2} + A_4 y_{j0} y_{j1}' y_{j2}' + A_5 y_{j0} y_{j1}' y_{j2} \\ & + A_6 y_{j0} y_{j1} y_{j2}' + A_7 y_{j0} y_{j1} y_{j2}, \end{aligned} \tag{3}$$

Our general problem, then, is to find $R_s=E[x_jx_{j+s}]$ (the mean of x_jx_{j+s}) with the set Φ as a parameter. Given this, we can obtain the normalized correlation ρ_s, defined as

$$\rho_s = \frac{R_s - E[x_j]E[x_{j+s}]}{R_0 - E[x_j]E[x_{j+s}]} = \frac{R_s - (\bar{x})^2}{R_0 - (\bar{x})^2}. \tag{4}$$

Since in our problem the location of the start of each period is random, X may be considered as narrow-sense stationary, so that neither $\bar{x}=E[x_j]=E[x_{j+s}]$ nor R_s depend on j.

Let us assume that all the periods $p_0, p_1, \cdots, p_k, \cdots, p_{n-1}$ of $Y_0, Y_1, \cdots, Y_k \cdots, Y_{n-1}$, respectively, are relative prime numbers. Then, R_s is periodic with period no greater than

$$\prod_{k=0}^{n-1} p_k$$

and we need consider only the interval

$$0 < s < \prod_{k=0}^{n-1} p_k.$$

A singular point s^* of this interval is defined as a number which can be divided by any p_k; *i.e.*, it is a product of any combination of the periods. There are 2^n-1 singular points of interest in our interval. Furthermore, our independence assumption causes ρ_s to be zero when $s \neq s^*$.

It is convenient to introduce a binary number B which corresponds to s^*. Thus, when $B=01101$, then $s^*=p_1p_2p_4$. Formally we can write,

$$s^* = \prod_{k=0}^{n-1} p_k^{b_k}. \tag{5}$$

b_k is either 1 or 0 depending on whether s^* is divisible by p_k, so that the binary number $B=b_0b_1b_2 \cdots b_{n-1}$ specifies s^*.

We will now outline a procedure which yields a general formulation for ρ_s. We thought it advisable to first illustrate the method for the case $n=3$ before working out the general equations. When $q=\frac{1}{2}$, the results will be particularly simple.

Table I shows all values of s^* for $n=3$, the corresponding values of B, and the conditions which apply in each case. The abbreviation "ind." means "independent of."

TABLE I

s^*	B	Conditions		
p_0	100	$y_{j0}=y_{j+s^*,0}$	y_{j1} ind. $y_{j+s^*,1}$	y_{j2} ind. $y_{j+s^*,2}$
p_1	010	y_{j0} ind. $y_{j+s^*,0}$	$y_{j1}=y_{j+s,1}$	y_{j2} ind. $y_{j+s^*,2}$
p_2	001	y_{j0} ind. $y_{j+s^*,0}$	y_{j1} ind. $y_{j+s^*,1}$	$y_{j2}=y_{j+s^*,2}$
p_0p_1	110	$y_{j0}=y_{j+s^*,0}$	$y_{j1}=y_{j+s^*,1}$	y_{j2} ind. $y_{j+s^*,2}$
p_0p_2	101	$y_{j0}=y_{j+s^*,0}$	y_{j1} ind. $y_{j+s^*,1}$	$y_{j2}=y_{j+s^*,2}$
p_1p_2	011	y_{j0} ind. $y_{j+s^*,0}$	$y_{j1}=y_{j+s^*,1}$	$y_{j2}=y_{j+s^*,2}$
$p_0p_1p_2$	111	$y_{j0}=y_{j+s^*,0}$	$y_{j1}=y_{j+s^*,1}$	$y_{j2}=y_{j+s^*,2}$

The heart of our procedure is the separation of the variables satisfying the different conditions. We factor the variables which satisfy the "equals" condition. Thus, for $B=100$, (3) becomes

$$x_j = y_{j0}'[A_0y_{j1}'y_{j2}' + A_1y_{j1}'y_{j2} + A_2y_{j1}y_{j2}' + A_3y_{j1}y_{j2}] + y_{j0}[A_4y_{j1}'y_{j2}' + A_5y_{j1}'y_{j2} + A_6y_{j1}y_{j2}' + A_7y_{j1}y_{j2}]. \quad (6)$$

For $B=011$, (3) becomes

$$x_j = y_{j1}'y_{j2}'[A_0y_{j0}' + A_4y_{j0}] + y_{j1}'y_{j2}[A_1y_{j0}' + A_5y_{j0}] + y_{j1}y_{j2}'[A_2y_{j0}' + A_6y_{j0}] + y_{j1}y_{j2}[A_3y_{j0}' + A_7y_{j0}]. \quad (7)$$

Factoring for other values of B is left to the reader.

Once the factoring is done, evaluation of R_s is greatly simplified. For a given B and s^*, x_j and x_{j+s^*} are both factored in the same way. Bracketed terms of x_j are then independent of bracketed terms of x_{j+s}. Because of the terms outside the brackets, all cross products vanish. Thus, for $B=100$,

$$R_s = E[y_{j0}']\{E[A_0y_{j1}'y_{j2}' + A_1y_{j1}'y_{j2} + A_2y_{j1}y_{j2}' + A_3y_{j1}y_{j2}]\}^2 + E[y_{j0}]\{E[A_4y_{j1}'y_{j2}' + A_5y_{j1}'y_{j2} + A_6y_{j1}y_{j2}' + A_7y_{j1}y_{j2}]\}^2. \quad (8)$$

For $B=011$ [with the empty brackets signifiying the means of the filled brackets of (7)],

$$R_s = E[y_{j1}'y_{j2}']\{\ \}^2 + E[y^j{}_1'y_{j2}]\{\ \}^2 + E[y_{j1}y_{j2}']\{\ \}^2 + E[y_{j1}y_{j2}]\{\ \}^2. \quad (9)$$

Since the probability that $y_{jk}=1$ is q, it follows that $E[y_{jk}]=q$ and $E[y_{jk}']=1-q$. Then (8) and (9) become [letting $R(p_0)=R_s$ for $s=p_0$ and $R(p_1p_2)=R_s$ for $s=p_1p_2$]

$$R(p_0) = (1-q)[A_0(1-q)^2+(A_1+A_2)q(1-q)+A_3q^2] + q[A_4(1-q^2)+(A_5+A_6)q(1-q)+A_7q^2]. \quad (10)$$

$$R(p_1p_2) = (1-q)^2[A_0(1-q)+A_4q] + q(1-q)[(A_1+A_2)(1-q)+(A_5+A_6)q] + q^2[A_3(1-q)+A_7q]. \quad (11)$$

In the same way R_s may be obtained for all $s=s^*$.

In general, x_j can be factored by the following procedure: Let w_{j0} be the first y_{jk} for which $b_k=1$. Let w_{j1} be the second such value, etc. For example, if B is

$$b_0b_1b_2b_3b_4b_5b_6b_7b_8b_9,$$
$$0\,0\,1\,0\,1\,1\,0\,0\,1\,1,$$

then

$$w_{j0}=y_{j2},\ w_{j1}=y_{j4},\ w_{j2}=y_{j5},\ w_{j3}=y_{j8},\ w_{j4}=y_{j9}.$$

Similarly, let v_{j0} be the first y_{jk} for which $b_k=0$, etc. Then

$$v_{j0}=y_{j0},\ y_{j1}=y_{j1},\ v_{j2}=y_{j3},\ v_{j3}=y_{j6},\ v_{j5}=v_{j7}.$$

Clearly, H_{ji} can be factored into W_{jr} and V_{jt}, where the latter two are products of the w's and v's, and where r and t have the same meaning as i for H_{ji}. For any H_{ji}, we can write (where B determines the appropriate r and t)

$$H_{ji} = W_{jr}V_{jt}.$$

For example, using the same B as before,

$$H_{ji} = y_{j0}'y_{j1}'y_{j2}'y_{j3}'y_{j4}'y_{j5}'y_{j6}'y_{j7}y_{j8}y_{j9}' = [y_{j2}'y_{j4}'y_{j5}'y_{j8}y_{j9}'] \times [y_{j0}'y_{j1}'y_{j3}'y_{j6}'y_{j7}] = W_{jr}V_{jt}.$$

Eq. (1) may now be written

$$x_j = \sum_{r=0}^{2^m-1} W_{jr}\left[\sum_{t=0}^{2^{n-m}-1} A_{rt}{}^*V_{jt}\right], \quad (12)$$

and

$$x_{j+s^*} = \sum_{r=0}^{2^m-1} W_{jr}\left[\sum_{t=0}^{2^{n-m}-1} A_{rt}{}^*V_{j+s^*,t}\right], \quad (13)$$

where m is the number of w's in any W; thus, $n-m$ is the number of v's in any V.

$A_{rt}{}^*$ can be related to A_i by defining subsets T_r of Φ. There are 2^m such subsets, each containing 2^{n-m} different $A_{rt}{}^*$. For example, in (7) $A_{00}{}^*=A_0$, $A_{01}=A_4$ belong to T_0, $A_{10}{}^*=A_1$, $A_{11}{}^*=A_5$ belong to T_1, $A_{20}{}^*=A_2$, $A_{21}{}^*=A_6$ belong to T_2 and $A_{30}{}^*=A_3$, $A_{31}{}^*=A_7$ belong to T_3.

In Section III we shall show how the subsets T_r can be constructed so that every $A_{rt}{}^*$ can be associated with the appropriate A_i. In this section, we shall obtain R_s in terms of the $A_{rt}{}^*$.

We note that W_{jr}, V_{jt} and $V_{j+s^*,t}$ are independent. Then, taking the mean value of the product of (12) and (13), we obtain

$$R_s = \sum_{r=0}^{2^m-1} E[W_{jr}]\left\{\sum_{t=0}^{2^{n-m}-1} A_{rt}{}^*E[V_{jt}]\right\}^2. \quad (14)$$

When $q=\frac{1}{2}$, $E[W_{jr}]=(\frac{1}{2})^m$ for all r and j, and $E[V_{jt}]=(\frac{1}{2})^{n-m}$ for all t and j. Then, letting

$$\alpha_r = \sum_{t=0}^{2^{n-m}-1} A_{rt}{}^*,$$

(14) reduces to

$$R_s = (\tfrac{1}{2})^{2n-m}\sum_{r=0}^{2^m-1}\alpha_r{}^2 \quad (15)$$

Thus, for $q=\frac{1}{2}$, R_s depends only on the *sums* of A_i over each of the subsets T_r.

An interesting form is obtained for the numerator of ρ_s by noting that $\bar{x}$ at $q=\frac{1}{2}$ can be written as

$$\bar{x} = \left(\frac{1}{2}\right)^n\left[\sum_{r=0}^{2^m-1}\alpha_r\right]. \quad (16)$$

The difference between (15) and the square of (16) is factorable and reduces to

$$[R_s - \bar{x}^2]2^{2n} = \sum_{\substack{r,t=0\\ r\neq t}}^{2^m-1}(\alpha_r-\alpha_t)^2. \quad (17)$$

To illustrate the symbolism, let $n=5$, $m=2$. Then, from (15) and (16):

$$[R_s - (\bar{x})^2]2^5 = 4[\alpha_0^2 + \alpha_1^2 + \alpha_2^2 + \alpha_3^2] - [\alpha_0 + \alpha_1 + \alpha_2 + \alpha_3]^2 = (\alpha_0 - \alpha_1)^2 + (\alpha_0 + \alpha_2)^2 + (\alpha_0 - \alpha_3)^2 + (\alpha_1 - \alpha_2)^2 + (\alpha_1 - \alpha_3)^2 + (\alpha_2 - \alpha_3)^2.$$

The right side of (18) is the expansion of (17).

III. Subdivision of Φ into the Subsets T_r

It is clear from (17) that ρ_s depends only on how the set Φ is subdivided. Furthermore, only s^* influences this subdivision. In this section, we shall show how this division is effected for a particular case. The method is perfectly general and applicable to any other case.

Let $n=7$ and $s^*=p_2p_5p_6$ so that $B=0\,0\,1\,0\,0\,1\,1$. Now construct a 7-stage binary counter, as shown in Fig. 1.

Note that the top counter corresponds to the zeros of B and the bottom counter to the ones of B. If, for example, counter 1 is in state 0 0 1 1 and counter 2 is in the state 1 1 0, then the combined state is 0 0 1 1 1 1 0, obtained by interleaving the two counters.

Now,

1) r, the subscript of T_r, is the count of counter 2.
2) the subscript of all A_i's belonging to a given T_r are generated by going through a complete cycle of counter 1 and, for every state in counter 1, obtaining the count corresponding to the combined states of counters 1 and 2.

Thus, to generate all the A_i's in T_0, set counter 2 to 0 0 0. The successive combined counts obtained by cycling counter 1 are:

0, 4, 8, 12, 32, 36, 40, 44, 64, 68, 72, 76, 96, 100, 104, 108.

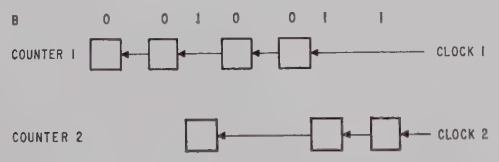


Fig. 1—Binary counter for generating subsets T_r.

Thus

$$A_{00}^* = A_0, \qquad A_{01}^* = A_4, \qquad A_{02}^* = A_8, \text{ etc.}$$

To construct T_5, set counter 2 to 1 0 1. This adds a constant (seventeen) to the combined count. Thus, the subscripts for T_5 can be generated by adding 17 to each subscript from T_0.

For this example, $m=3$, so there are 8 subsets of Φ. Table II shows the 8 corresponding values of α_r. C_r is the count of counter 2 for a given T_r. No two combined states are ever equal; thus all sets T_r are nonoverlapping.

IV. Proof That Only Addition Modulo 2 (and the Complementary Set) Yields a Noise-Like Correlation Function [for $q=\frac{1}{2}$]

How do the correlation functions generated by our pseudo-random sequences compare with those of true noise? One model for purely random noise is that of an infinite sequence of independent variables. The ρ_s for this true noise is unity for $s=0$ and 0 elsewhere.

Now, it has been shown[1] that a particular set Φ^* of A_i gives a ρ_s which is unity for

$$s = K \prod_{k=0}^{n-1} p_k,$$

where $K=0, 1, 2, \cdots$. Thus, at least over the maximum period of X, the correlation function of X is like that of pure noise. (The set Φ^* defines addition modulo 2.)

Φ^* is constructed by setting $A_i=1$ whenever H_{ji} contains an odd number of uncomplemented Y_{jk}'s and $A_i=0$ otherwise. The complement of Φ^*, obtained by complementing all A_i in Φ^*, also gives a ρ_s like pure noise.

In this section, we show that Φ^* and its complement are the *only* two sets that give this noiselike ρ_s. First we consider n specific values of s^*, given by

$$s^* = \frac{\prod_{k=0}^{n-1} p_k}{p_0}, \frac{\prod_{k=0}^{n-1} p_k}{p_1}, \frac{\prod_{k=0}^{n-1} p_k}{p_2}, \cdots, \frac{\prod_{k=0}^{n-1} p_k}{p_{n-1}}. \quad (19)$$

TABLE II

T_r	C_r	
0	0	$\alpha_0=A_0+A_4+A_8+A_{12}+A_{32}+A_{36}+A_{40}+A_{44}+A_{64}+A_{68}+A_{72}+A_{76}+A_{96}+A_{100}+A_{104}+A_{108}$
1	1	$\alpha_1=A_1+A_5+A_9+A_{13}+A_{33}+A_{37}+A_{41}+A_{45}+A_{65}+A_{69}+A_{73}+A_{77}+A_{97}+A_{101}+A_{105}+A_{109}$
2	2	$\alpha_2=A_2+A_6+A_{10}+A_{14}+A_{34}+A_{38}+A_{42}+A_{46}+A_{66}+A_{70}+A_{74}+A_{78}+A_{98}+A_{102}+A_{106}+A_{110}$
3	3	$\alpha_3=A_3+A_7+A_{11}+A_{15}+A_{35}+A_{39}+A_{43}+A_{47}+A_{67}+A_{71}+A_{75}+A_{79}+A_{99}+A_{103}+A_{107}+A_{111}$
4	16	$\alpha_4=A_{16}+A_{20}+A_{24}+A_{28}+A_{48}+A_{52}+A_{56}+A_{60}+A_{80}+A_{84}+A_{88}+A_{92}+A_{112}+A_{116}+A_{120}+A_{124}$
5	17	$\alpha_5=A_{17}+A_{21}+A_{25}+A_{29}+A_{49}+A_{53}+A_{57}+A_{61}+A_{81}+A_{85}+A_{89}+A_{93}+A_{113}+A_{117}+A_{121}+A_{125}$
6	18	$\alpha_6=A_{18}+A_{22}+A_{26}+A_{30}+A_{50}+A_{54}+A_{58}+A_{62}+A_{82}+A_{86}+A_{90}+A_{94}+A_{114}+A_{118}+A_{122}+A_{126}$
7	19	$\alpha_7=A_{19}+A_{23}+A_{27}+A_{31}+A_{51}+A_{55}+A_{59}+A_{63}+A_{83}+A_{87}+A_{91}+A_{95}+A_{115}+A_{119}+A_{123}+A_{127}$

For each s^* specified by (19), there is an arrangement of the type shown in Fig. 1. For example, when

$$s^* = \frac{\prod_{k=0}^{n-1} p_k}{p_2},$$

we obtain Fig. 2.

$n/2$ subsets, each containing 2 of the A_i, are generated from Fig. 2. T_0 contains A_0 and $A_{2^{n-3}}$, T_1 contains A_1 and $A_{2^{n-3}+1} \cdots$; T_r contains A_r and $A_{2^{n-3}+r}$, that is $\alpha_r = A_r + A_{2^{n-3}+r}$.

Now let us find which of the sets Φ satisfy the equation ρ_{s^*} for all s^* specified by (19). From (17) we immediately see that $\alpha_r = \alpha_t$ for all r and t. Since each set T_r contains only two A_i, the only possible values of α_r are 0, 1 and 2. But $\alpha_r = 0$ or $\alpha_r = 2$ are trivial cases, since they yield the results $A_i = 0$ for all i, or $A_i = 1$ for all i. Thus we are left with the equation $\alpha_r = 1$ for all r.

Returning to Fig. 2, we see that 2^{n-1} equations are generated, of the form

$$A_r + A_{2^{n-3}+r} = 1, \qquad r = 0, 1, 2, \cdots, 2^{n-1}. \quad (20)$$

n such groups of equations are produced, one group for each s^* specified by (19). Thus there are a total of $n2^{n-1}$ linear equations in the A_i, more than are needed to uniquely specify all the A_i. More careful study shows, however, that we have exactly $2^n - 1$ independent equations whose *only* solutions give us Φ^* and its complement.

We have shown that *only* Φ^* (and its complement) gives $\rho_{s^*} = 0$ for n specific values of s^*. It therefore follows immediately that *only* Φ^* ((and its complement) give $\rho_{s^*} = 0$ for all s^* (except at the periods of X).

As an example, if $n = 4$, the set Φ^* contains

$$A_0 = A_3 = A_5 = A_9 = A_6 = A_{10} = A_{12} = A_{15} = 0,$$
$$A_i = 1 \text{ for all other } i = 0, 1, \cdots, 15.$$

V. The Correlation Function for a Set of Random $A_i[q = \frac{1}{2}]$

Assume that all A_i are picked at random and are independent, and that, for each i, the probability of $A_i = 1$ is $\frac{1}{2}$.

Factoring (1) by the method used in Section II, we can obtain

$$x_j = \sum_{r=0}^{2^m-1} W_{jr} \sum_{t=0}^{2^{n-m}-1} A_{rt}{}^* V_{jt}. \quad (21)$$

Noting that $W_{jr_1} W_{jr_2} = 0$ for $r_1 \neq r_2$, and that $W_{jr}^2 = W_{jr}$, we obtain

$$x_j x_{j+s} = \sum_{r=0}^{2^m-1} W_{jr} \left[\sum_{t=0}^{2^{n-m}-1} A_{rt}{}^* V_{jt} \right] \cdot \left[\sum_{t=0}^{2^{n-m}-1} A_{rt}{}^* V_{j+s,t} \right]. \quad (22)$$

Consider any cross-product term of the two rightmost sums. It may be written, $A_{rt_1}{}^* A_{rt_2}{}^* V_{jt_1} V_{j+s,t_2}$. When $t_1 = t_2$,

$$E[A_{rt_1}{}^* A_{rt_2}{}^*] = \tfrac{1}{2}; \quad \text{when} \quad t_1 \neq t_2, \quad E[A_{rt_1}{}^* A_{rt_2}{}^*] = \tfrac{1}{4}.$$

Furthermore,

$$E[V_{jt_1} V_{j+s,t_2}] = (\tfrac{1}{2})^{2(n-m)},$$

always. Since there are 2^{n-m} terms in (22) for which $t_1 = t_2$ and $2^{2(n-m)} - 2^{n-m}$ terms for which $t_1 \neq t_2$, the mean of (22) becomes

$$E[x_j x_{j+s}] = \sum_{r=0}^{2^m-1} E[W_{jr}] \{ 2^{n-m} [\tfrac{1}{2} (\tfrac{1}{2})^{2(n-m)}] + (2^{2(n-m)} - 2^{n-m}) [\tfrac{1}{4} (\tfrac{1}{2})^{2(n-m)}] \}. \quad (23)$$

Since $E[W_{jr}] = (\tfrac{1}{2})^m$, (23) reduces to

$$R_s = \tfrac{1}{4} + \tfrac{1}{4} (\tfrac{1}{2})^{n-m}. \quad (24)$$

As before, (24) holds only for the singular points $s = s^*$, and ρ_s is zero elsewhere. Since $R_0 = \frac{1}{2}$ and $\bar{x} = \frac{1}{2}$ we get

$$\rho_s = (\tfrac{1}{2})^{n-m}. \quad (25)$$

Fig. 3 shows a graph of ρ_s vs s for the special case $n = 4$ and $p_0 = 3$, $p_1 = 4$, $p_2 = 5$, $p_3 = 7$. Points corresponding to multiples of any combinations of periods have been omitted.

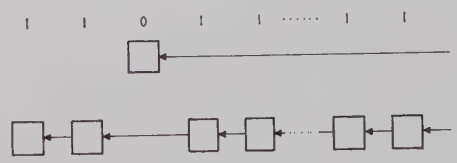


Fig. 2—Binary counter for

$$s^* = \frac{\prod_{k=0}^{n-1} p_k}{p_2}.$$

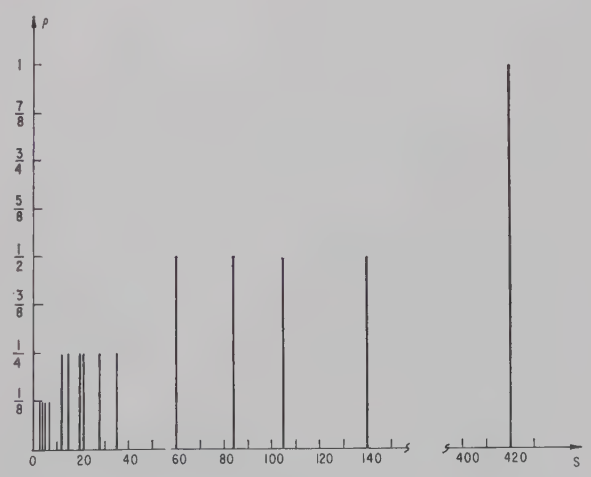


Fig. 3—The correlation function for a random set of $A_i (n = 4, q = \frac{1}{2})$.

V. Degenerate Sequences

Up to now we have assumed that all the periods p_k are relative prime numbers and, furthermore, that this limits the period of X to be the product of all p_k. For certain sets Φ it is possible for the period of X to be less than

$$\prod_{k=0}^{n-1} p_k.$$

Such sequences we will call degenerate and, in this section, we will find how many of the 2^{2^n} possible Φ's produce degenerate sequences.

We also define the degree of degeneration as follows: if X is a function of n sequences $Y_0, Y_1, \cdots, Y_{n-1}$, but if the period of X for a given Φ is the product of any h of the n periods, we say that X is degenerate of degree h.

Let $N_n(h)$ be the number of different Φ's which cause degeneration of degree h for n sequences. Then we have two basic relations:

$$N_n(n) = 2^{2^n} - \sum_{h=0}^{n-1} N_n(h), \tag{26}$$

$$N_n(h) = \frac{n!}{h!(n-h!)} N_h(h) = \binom{n}{h} N_h(h). \tag{27}$$

From (26) and (27) we can derive the explicit result

$$N_n(h) = \binom{n}{h} \begin{vmatrix} \binom{0}{0} & 0 & 0 & 0 \cdots 0 & 2^{2^0} \\ \binom{1}{0} & \binom{1}{1} & 0 & 0 \cdots 0 & 2^{2^1} \\ \binom{2}{0} & \binom{2}{1} & \binom{2}{2} & 0 \cdots 0 & \cdot \\ \binom{3}{0} & \binom{3}{1} & \binom{3}{2} & \binom{3}{3} \cdots 0 & \cdot \\ \binom{4}{0} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \binom{n}{0} & \binom{n}{1} & \binom{n}{2} & \cdots & \binom{n}{n-1} \; 2^{2^n} \end{vmatrix}. \tag{28}$$

From (28) or by iteration from (26) and (27) we obtain some numerical values:

$$N_n(0) = 2, \quad N_n(1) = 2n, \quad N_n(2) = \frac{10n(n-1)}{2},$$

$$N_n(3) = \frac{218n(n-1)(n-2)}{3!},$$

$$N_n(4) = \frac{64{,}594n(n-1)(n-2)(n-3)}{4!}.$$

For $h \geq 6$ the percentage of degenerate truth tables becomes very small.

VI. Discussion

A very general form of a binary digit generator has been proposed which is capable of producing a large number of pseudo-random output sequences. To study and classify these output sequences, a general procedure for evaluating the mean correlation function has been found which provides insight into some of the properties of these sequences. This procedure, together with several of the examples, shows the following:

1) The normalized mean-correlation function may vary with different truth tables.
2) When the digits of the constituent periodic sequences are 1 or 0 with equal probability, then only addition modulo 2 (and its complementary set) produce a normalized mean correlation comparable to sequences of truly random digits.
3) When truth tables are chosen at random, the normalized mean correlation of the output sequences will increase with delay s. With several long periodic sequences, the normalized mean correlation will be quite small over a restricted range of s.

An investigation into Boolean functions which always produce sequences whose periods are less than

$$\prod_{k=0}^{n-1} p_k$$

gave an explicit result for the number of these functions. For many periodic sequences ($n \geq 6$) a very small percentage of the truth tables are degenerate.

Several interesting problem areas remain to be investigated for this general form of digit generator. One such problem area deals with finding a suitable mathematical description of the digit generator to give precise information about the number and types of output sequences available.[4] A second problem area deals with choosing the probability of 1's and 0's in the periodic sequences and also choosing truth tables to provide many output sequences with the desired mean frequency of occurrences[5] of 1's or 0's as well as a specified normalized mean correlation. A third problem area deals with finding a general procedure which yields the dispersion about the mean correlation for the different truth tables.[6] The dispersion will yield an estimate of the closeness of the output sequences to the mean.

[4] The authors have observed many interesting properties of these sequences, but purposely omitted them since they did not have general descriptions of these sequences.

[5] The computation of the average occurrence of 1's in the output sequences was purposely omitted being relatively straightforward.

[6] Maximon obtained an expression for the dispersion for the truth table addition modulo 2.

Signed-Digit Numbe Representations for Fast Parallel Arithmetic*

ALGIRDAS AVIZIENIS†, MEMBER, IRE

Summary—This paper describes a class of number representations which are called *signed-digit* representations. Signed-digit representations limit carry-propagation to one position to the left during the operations of addition and subtraction in digital computers. Carry-propagation chains are eliminated by the use of redundant representations for the operands. Redundancy in the number representation allows a method of fast addition and subtraction in which each sum (or difference) digit is the function only of the digits in two adjacent digital positions of the operands. The addition time for signed-digit numbers of any length is equal to the addition time for two digits. The paper discusses the properties of signed-digit representations and arithmetic operations with signed-digit numbers: addition, subtraction, multiplication, division and roundoff. A brief discussion of logical design problems for a signed-digit adder concludes the presentation.

I. Introduction

THIS PAPER describes a class of number representations which are called *signed-digit* representations. Signed-digit representations limit carry propagation to one position to the left during the operations of addition and subtraction in digital computers. Carry-propagation chains are eliminated by the use of redundant representations for the operands. In a *conventional number representation* with an integer radix $r>1$ each digit is allowed to assume exactly r values: $0, 1, \cdots, r-1$. In a *redundant representation* with the same radix r each digit is allowed to assume more than r values.

Previous methods of carry elimination formed redundant representations by the combination of explicitly identified digits, such as stored carries or borrows, with conventional number representations.[1–6] A difficulty in redundant representations of the stored-carry type is the lack of unique representation for the zero algebraic value of a number; the former sign digit (now indicator digit) does not always indicate the true sign of a number and overflow detection becomes more complicated. Furthermore, the number of states used to represent the values of one digital position is doubled by the binary stored-carry digit; this amount of redundancy is excessive.

In the method described here, each digit of a positional constant radix number representation with an integer radix r is allowed to assume q values

$$r+2 \leq q \leq 2r-1,$$

that is, more than the r values allowed in the conventional representation.[7] Both positive and negative digit values are allowed for this purpose. Redundancy in the number representation allows a method of fast addition and subtraction in which each sum (or difference) digit is the function only of the digits in two adjacent digital positions of the operands. These operations are called *totally-parallel* addition and subtraction. The requirement for totally-parallel addition and subtraction determines the minimum redundancy ($r+2$ values) which is necessary in the representation of one digit. The upper limit for the redundancy of digit values results from the requirement for a unique representation of the zero algebraic value of a number. To satisfy this requirement the magnitude of allowed digit values may not exceed $r-1$.

Requirements of totally-parallel addition and subtraction and of a unique representation for the zero value are satisfied by a class of redundant representations for radices $r>2$ which are called *signed-digit* representations. The digits of a signed-digit representation individually assume both positive and negative integer values and contain the sign information for the number; no special sign digit is necessary. This property leads to the name of "signed-digit" representations. The number of digit values in a radix $r>2$ representation ranges from a required minimum of $r+2$ to an allowable maximum of $2r-1$.

The following sections of this paper describe the properties of individual digits and of complete numbers in signed-digit representations. After development of these properties, arithmetic operations for signed-digit representations are discussed. Examples are given of

* Received by the PGEC, March 23, 1961; revised manuscript received, May 22, 1961.

† Jet Propulsion Lab., California Inst. Tech., Pasadena, Calif.

[1] A. W. Burks, *et al.*, "Preliminary Discussion of the Logical Design of an Electronic Computing Instrument," Institute for Advanced Study, Princeton, N. J.; pt. 1, vol. I, 2nd ed. revised; September, 1947.

[2] J. E. Robertson, "Preliminary Design of an Arithmetic Unit for Use with a Self-Checking Binary Parallel Digital Computer," Digital Computer Lab., University of Illinois, Urbana, Ill., Internal Rept. No. 19; June, 1950.

[3] Staff of the Digital Computer Lab., "On the Design of a Very High-Speed Computer," University of Illinois, Urbana, Ill., Rept. No. 80, ch. 8; October, 1957.

[4] G. Metze, "A Study of Parallel One's Complement Arithmetic Units with Separate Carry or Borrow Storage," Ph.D. dissertation, University of Illinois, Urbana, Ill., 1958; Digital Computer Lab., University of Illinois, Rept. No. 81; November 11, 1957.

[5] G. Metze and J. E. Robertson, "Elimination of carry propagation in digital computers," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France, June 15–20, 1959; pp. 389–396.

[6] J. E. Robertson, "Redundant number systems for digital computer arithmetic," notes for the University of Michigan Engrg. Summer Conf., "Topics in the Design of Digital Computing Machines," Ann Arbor, Mich.; July 6–10, 1959.

[7] A. Avizienis, "A Study of Redundant Number Representations for Parallel Digital Computers," Ph.D. dissertation, University of Illinois, Urbana, Ill., 1960; Digital Computer Laboratory, University of Illinois, Rept. No. 101; May 20, 1960.

signed-digit addition, subtraction, multiplication and division. The paper concludes with some logical design considerations for a signed-digit adder.

II. Properties of Signed-Digit Representations

In this section the class of signed-digit representations is derived according to four requirements which are postulated as necessary for number representations in fast parallel arithmetic.

A. Requirements for Signed-Digit Representations

The purpose of signed-digit representations is to allow addition and subtraction of two numbers in which no serial signal propagation is required along the adder; that is, the time duration of the operation is independent of the length of the operands and is equal to the time required for the addition or subtraction of two digits. Furthermore, the signed-digit representations must have a unique representation of zero algebraic value of a number. The desired principal properties of signed-digit representations are specified by the following description of a signed-digit number:

A *signed-digit number* is represented by $n+m+1$ digits z_i $(i=-n, \cdots, -1, 0, 1, \cdots, m)$ and has the algebraic value

$$Z = \sum_{i=-n}^{m} z_i r^{-i},$$

where the values of r and z_i are such that the following requirements are satisfied:

1) The radix r is a positive integer.
2) The algebraic value $Z=0$ has a unique representation.
3) There exist transformations between the conventional representation and the signed-digit representation for every algebraic value Z within a specified range.
4) Totally-parallel addition and subtraction is possible for all digits in corresponding positions of two representations.

The arithmetic operations of totally-parallel addition and subtraction of two digits z_i and y_i from the corresponding ith positions of the representations of numbers Z and Y are defined as follows:

Definition 1: Addition of digits z_i and y_i is totally-parallel if the following two conditions are satisfied:

1) The *sum digit* s_i (ith digit of the sum $S=Z+Y$) is a function only of the *augend digit* z_i, *addend digit* y_i and the *transfer*[8] *digit* t_i from the $(i+1)$th position on the right: $s_i=f(z_i, y_i, t_i)$.

[8] The term "transfer digit" is used here instead of the commonly used terms "carry" or "borrow" for two reasons:

1) the transfer digit may assume both positive and negative values in either addition or subtraction;
2) unlike the "carry" or "borrow" of conventional addition or subtraction, the transfer digit is never propagated past the first adder position on the left.

2) The transfer digit t_{i-1} to the $(i-1)$th position on the left is a function only of the augend digit z_i and the addend digit y_i: $t_{i-1}=f(z_i, y_i)$.

Definition 2: Totally-parallel subtraction of the *subtrahend digit* y_i from the *minuend digit* z_i is performed as the totally-parallel addition of the additive inverse of y_i, *i.e.*, $z_i-y_i=z_i+(-y_i)$.

Definitions 1 and 2 impose limiting conditions on the values which the digits z_i and y_i may assume. The addition of two digits is performed in two successive steps. First, an outgoing transfer digit t_{i-1} and an *interim sum digit* w_i are formed:

$$z_i + y_i = rt_{i-1} + w_i. \tag{1}$$

Then the sum digit s_i is formed:

$$s_i = w_i + t_i. \tag{2}$$

Definition 1 will be satisfied if the range of values which s_i may assume in (2) does not exceed the allowed range of values for the digits z_i and y_i in (1). The block diagram of a totally-parallel adder for signed-digit representations is shown in Fig. 1.

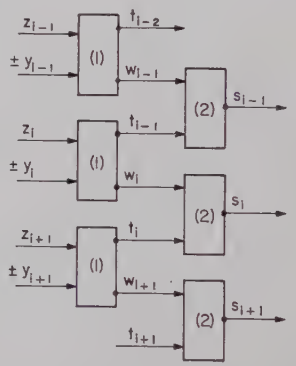


Fig. 1—Section of a totally-parallel adder for signed-digit representation.

Definition 2 will be satisfied if, for every allowed nonzero value of the digit y_i, there exists an additive inverse in the set of all allowed values of y_i;

$$\text{for every } y_i=a, \text{ there exists } y_i=-a \text{ such that } a+(-a)=0. \tag{3}$$

The requirement for unique representation of the zero value of a number will be satisfied by the condition

$$|z_i| \leq r-1. \tag{4}$$

The requirement for conversion from the conventional sign and magnitude, m-digit representation to signed-digit representation (m digits long), will be satisfied if the procedure of totally-parallel addition, applied to the nonsign digits x_i of the conventional representation,

$$x_i = rt_{i-1} + w_i \tag{5}$$

$$z_i = w_i + t_i, \tag{6}$$

will yield an allowed value of digit z_i for every positive and negative value of the digit x_i ($x_i = 0, 1, \cdots, r-1$) in the conventional radix r representation. Conversion from signed-digit to conventional representation is always possible, since condition (4) allows the interpretation of any signed-digit representation as the sum of two conventional sign and magnitude representations, one consisting of the positive and one of the negative digits of the signed-digit representation.

B. Required and Allowed Values of Digits

The conditions (1)–(6) of the preceding section determine the required and allowed values of one digit z_i for an arbitrary integer radix $r \geq 2$. As the first step, the required and allowed values of t_i and w_i in (1) and (2) must be established. The conditions (1)–(4) establish[9] the set of values

$$t_i = -1, 0, 1 \tag{7}$$

as the least sufficient set of values for the transfer digit t_i, and the condition

$$|w_i| \leq r - 2 \tag{8}$$

as the upper limit for the magnitude of the interim sum w_i if t_i is limited by (7). Values of $|t_i| > 1$ have no practical application for two-digit operations, but they are necessary for simultaneous totally-parallel addition and subtraction of several digits which will be discussed later. An immediate result of (8) is the restriction of allowed values of radix r to the integer values

$$r > 2 \tag{9}$$

because, for $r = 2$, the only allowed value of $w_i = 0$ does not satisfy (5) for the value $x_i = 1$.

Requirements (5), (7) and (8) establish that at least r values of w_i are required if all $2r-1$ positive and negative values of x_i are to be transformed according to (5), and that the values $w_i = -1, 0, 1$ should be included for all $r > 2$. The relationship between the greatest value $w_{\max}$ and smallest value $w_{\min}$ of w_i is:

$$w_{\max} - w_{\min} \geq r - 1, \tag{10}$$

where the equality sign applies if only r values of w_i are chosen. *The required set of values of* w_i therefore must consist of r integers, which include $-1, 0, 1$ and satisfy (8) and (5) for all $2r-1$ values of x_i. More than one such set may exist, but the choice of a sequence symmetric around zero is preferable in further development. *The set of all allowed values of* w_i is unique and consists of $2r-3$ integers from $-(r-2)$ to $r-2$.

The values of digits w_i and t_i and the totally-parallel subtraction requirement stated in (3) determine which values of z_i are required and how many values of z_i are allowed for a digit z_i of a radix $r \geq 3$ signed-digit representation. The digit z_i assumes the minimum number of values when the required set of w_i is chosen as a sequence of r integers $\{w_{\min}, \cdots, -1, 0, 1, \cdots, w_{\max}\}$ in which additive inverses exist for all (when r is odd) or all but one (when r is even) nonzero values of w_i. In both cases the required values of the sum digit s_i in (2) consist of a sequence of $r+2$ integers: $\{w_{\min}-1, w_{\min}, \cdots, -1, 0, 1, \cdots, w_{\max}, w_{\max}+1\}$.

For an odd radix $r_0 \geq 3$ we choose $w_{\max} = -w_{\min} = \frac{1}{2}(r_0 - 1)$; here the additive inverse exists for every value of w_i and s_i, and *the required (minimum) set of values for digits* z_i *or* y_i in (1) consists of the sequence of $r_0 + 2$ integers

$$\left\{-\tfrac{1}{2}(r_0+1), \cdots, -1, 0, 1, \cdots, \tfrac{1}{2}(r_0+1)\right\}. \tag{11}$$

For an even radix $r_e \geq 4$, either $(w_{\min}-1)$ or $(w_{\max}+1)$ in the set of the required values of s_i has no additive inverse. This additive inverse is required as a subtrahend input to the adder (y_i) in order to satisfy (3). *The required (minimum) set of values for the subtrahend digit* y_i consists of the sequence of $r_e + 3$ integers

$$\left\{-(\tfrac{1}{2}r_e+1), \cdots, -1, 0, 1, \cdots, \tfrac{1}{2}r_e+1\right\}. \tag{12}$$

The minimum set of values for the sum digit s_i and the augend (or minuend) digit z_i requires only $r_e + 2$ integers and either $\frac{1}{2}r_e + 1$ or $-(\frac{1}{2}r_e + 1)$ is omitted from (12) to give this set. Only $r_e + 2$ digit values require storage, but the adder must accept $r_e + 3$ values of the subtrahend digit y_i. For instance, given $r = 4$, the values $-2, -1, 0, 1, 2, 3$ are sufficient to represent any sum or difference, but the adder must also accept the subtrahend digit value $y_i = -3$ as the additive inverse of $y_i = 3$ during a subtraction.

C. Types of Signed-Digit Representations

The sets (11) and (12) are the required (minimum) sets of digit values which satisfy the requirements for signed-digit representations. They are the only allowed sets for radix 3 (values $-2, -1, 0, 1, 2$) and radix 4 (values $-3, -2, -1, 0, 1, 2, 3$). For all $r > 4$ there exists more than one set of allowed digit values. Since the maximum allowed absolute value of a digit is $r-1$ according to (4), all sequences of integers,

$$\{-a, -(a-1), \cdots, -1, 0, 1, \cdots, a-1, a\}$$
$$\tfrac{1}{2}(r_0+1) \leq a \leq r_0 - 1 \text{ or } \tfrac{1}{2}r_e + 1 \leq a \leq r_e - 1; \tag{13}$$
$$r_0 \text{ is an odd integer } r_0 \geq 3,$$
$$\text{and } r_e \text{ is an even integer } r_e \geq 4,$$

will satisfy the requirements for signed-digit representation. *The maximum allowed set of values* occurs when $a = r-1$ and consists of $2r-1$ integers. For instance, two sets exist for radix 5, one with 7 values (-3 to 3) and one with 9 values (-4 to 4). Four sets exist for radix 10, from 13 values (-6 to 6) to 19 values (-9 to 9).

All signed-digit representations, *i.e.*, representations which satisfy the requirements 1) to 4) of Section II-A, may now be described in terms of the allowed values of radix r and digits z_i as follows:

[9] Derivations of the results presented here appear in the Appendix

Definition 3: The *class of signed-digit representations* consists of all positional, constant radix representations of algebraic values

$$Z = \sum_{i=-n}^{m} z_i r^{-i}$$

in which the digits z_i assume one of the allowed sets of values given by (13) and the radix is a positive integer $r>2$. The *redundancy* of a signed-digit representation is *minimal* when $a=\frac{1}{2}(r_0+1)$ or $a=\frac{1}{2}r_e+1$, and the redundancy is *maximal* when $a=r_0-1$ or $a=r_e-1$ in (13).

A consequence of restriction (4), $|z_i| \leq r-1$, is an upper limit to the maximum weighted sum of all digits z_i $(i=k+1, \cdots, m)$ to the right of an arbitrary digit z_k in the signed-digit representation of the algebraic value Z, which is

$$\sum_{i=k+1}^{m} |z_i|_{\max} r^{-i} \leq r^{-k}(1 - r^{-m+k}) < r^{-k}. \quad (14)$$

Important consequences of (14) are as follows:

1) The sign of the algebraic value Z is indicated by the sign of the most significant nonzero digit.
2) The algebraic value Z is zero if, and only if, all digits of its signed-digit representation have the value $z_i=0$.
3) Given a signed-digit representation of the algebraic value Z, the signed-digit representation of $-Z$ is formed by changing the sign of the value of every nonzero digit z_i.

By repetitive formation and propagation of transfer digits, a signed-digit representation may be converted to a *canonical form* in which the values of all digits z_i are in the chosen set of the values of w_i; that is, no more transfer digits can be formed. A maximum of m additions of the number to zero may be required to put an m-digit number into the canonical form. If the set of values of w_i is the minimal (required) set, we obtain the *minimal canonical form*, which is nonredundant, since its digits assume only r values.

D. Range of Numbers and Overflow Indication

Generally, the range for algebraic values Z of fixed-point numbers (or mantissas in floating point arithmetic) is chosen as $1>Z>-1$ (or 1 or -1 may be included) in conventional number representations. Two considerations determine the choice of a convenient range for signed-digit representations: 1) the entire conventional range $1\geq Z\geq -1$ should be covered, and 2) a simple, immediate method of overflow indication should exist.

These two requirements will be satisfied if the absolute range limits for the algebraic value

$$Z = \sum_{i=0}^{m} z_i r^{-i}$$

of a signed-digit representation are defined to be

$$1 + |z_i|_{\max} \sum_{i=2}^{m} r^{-i} \geq Z \geq -1 - |z_i|_{\max} \sum_{i=2}^{m} r^{-i}. \quad (15)$$

Overflow is detected from the inspection of the two most significant digits z_0 and z_1 as follows:

positive overflow occurs if $z_0 = 1$ and

$$z_1 \geq 1, \text{ or if } z_0 > 1, \quad (16)$$

negative overflow occurs if $z_0 = -1$ and

$$z_1 \leq -1, \text{ or if } z_0 < -1. \quad (17)$$

In a redundant representation one algebraic value may be represented in more than one way. For this reason overflow will be indicated for some representations of numbers near the limits, but within the allowed maximum range, of (15). For the maximum allowed range of (15), and the overflow detection rules (16) and (17), there exist three ranges of overflow indication:

1) *Certain overflow indication*, irrespective of representation, will occur if Z exceeds the maximum range of (15)

$$|Z| > 1 + \frac{r^{-1} - r^{-m}}{r-1} |z_i|_{\max}. \quad (18)$$

2) *Potential overflow indication*, dependent on the specific representation of Z, covers the range

$$1 + \frac{r^{-1} - r^{-m}}{r-1} |z_i|_{\max} \geq |Z| \geq 1 + \frac{1}{r} - \frac{r^{-1} - r^{-m}}{r-1} |z_i|_{\max}. \quad (19)$$

3) *No overflow indication* for any representation of Z will occur in the range

$$|Z| < 1 + \frac{1}{r} - \frac{r^{-1} - r^{-m}}{r-1} |z_i|_{\max}. \quad (20)$$

The no-overflow range is minimum when $|z_i|_{\max} = r-1$. In this case, the range of (20) reduces to

$$|Z| < 1 + r^{-m}. \quad (21)$$

The range $1\geq Z\geq -1$ is therefore covered by every signed-digit representation and is considered as the allowed range for the scaling of numbers. It must be noted that an overflow indication may not occur until the algebraic value of a number reaches the certain overflow indication range of (18). Other ranges and more precise overflow detection may be chosen in an analogous manner.

E. Conversion Procedures

The conversion of a conventional (sign and magnitude) representation to a signed-digit representation of the same radix $r>2$ occurs according to (5) and (6).

Given the conventional representation of

$$Z = (1 - 2x_s) \sum_{i=-n}^{m} x_i r^{-i},$$

in which the digits x_i range over r values $0, 1, \cdots, r-1$, and $x_s=0$ if Z is positive and $x_s=1$ if Z is negative, the conversion to signed-digit representation is performed as follows:

1) Choose the set of allowed values ($w_{\min}, \cdots, -1, 0, 1, \cdots, w_{\max}$) of the interim sum w_i.
2) Interpret each digit x_i ($i=-n, \cdots, -1, 0, 1, \cdots, m$) to be negative if the sign digit $x_s=1$.
3) Consider each digit x_i as the sum of two digits in signed-digit representation and recode according to (5)

$$w_i = x_i - rt_{i-1},$$

where

$$t_{i-1} = 0 \text{ if } w_{\min} \leq x_i \leq w_{\max}$$
$$t_{i-1} = 1 \text{ if } x_i > w_{\max}$$
$$t_{i-1} = -1 \text{ if } x_i < w_{\min}.$$

4) Form the digits z_i of the signed-digit representation: $z_i=w_i+t_i$ for $i=-n, \cdots, -1, 0, 1, \cdots, m$, and $z_{-n-1}=t_{-n-1}$.

The conversion from signed-digit to a conventional representation may be performed by several methods. A direct approach is to consider the signed-digit number as the sum of two numbers in conventional representation of the same length, one of which is positive and the other negative. Negative and positive digits are separated to form these two numbers, which then can be added in an adder for conventional representations to yield the desired conventional representation. Conversion can also be executed in a serial manner, starting with the least significant digit of the signed-digit representation.[7]

F. Simultaneous Addition of Several Digits

Previous discussion of totally-parallel addition and subtraction was limited to simultaneous addition of two digits and therefore to two numbers only. However, rules (1) and (2) may be extended to simultaneous totally-parallel addition of an arbitrary number n of digits if values of the transfer digit $|t_i| > 1$ are allowed and the radix r is sufficiently large.

The conditions (1), (2), (4) and (10) yield the upper and lower limits for the greatest absolute value of t_i as follows:

$$\frac{n-1}{r-1} |z_i|_{\max} \leq |t_i|_{\max} \leq |z_i|_{\max} - |w_i|_{\max}. \quad (22)$$

The values of t_i are chosen as a sequence of integers ranging from $-|t_i|_{\max}$ to $|t_i|_{\max}$, and the values of w_i must satisfy (2) and (4); *i.e.*, $|w_i+t_i| \leq r-1$ for all values of t_i and w_i. Consequently, the number n of digits which may be added simultaneously is a function of the radix r and of the maximum allowed magnitudes of digits z_i and w_i

$$n \leq r - \frac{|w_i|_{\max}}{|z_i|_{\max}} (r-1). \quad (23)$$

If the greatest $|z_i|_{\max}$ and the least $|w_i|_{\max}$ (see Appendix) are used,

$$n \leq \tfrac{1}{2}(r_0 + 1) \quad \text{and} \quad n \leq \tfrac{1}{2}r_e \quad (24)$$

is the upper limit for n. For instance, a maximum of two digits may be added simultaneously for radix 4, three digits for radix 5, and five digits for radix 10.

A total of n numbers in signed-digit representation may be simultaneously added (or subtracted) in the totally-parallel mode and will yield a sum or difference in signed-digit representation if (22) and (23) are satisfied.

G. Modified Signed-Digit Representations

The requirement of totally-parallel addition and subtraction restricts signed-digit representations to radices $r>2$. Furthermore, at least $r+2$ values are required for the sum digit s_i in (2), and the subtraction requirement increases the required number of subtrahend digit values to $r+3$ for even radices, as shown in (12). The digit addition rules (1) and (2) may be modified to allow the propagation of the transfer digit over two digital positions to the left.[10] If this type of *two-transfer* addition is allowed, the radix $r=2$ may be used and only $r+1$ values are required for the sum digit. Two-transfer addition is executed in three successive steps (the adder diagram is shown in Fig. 2):

$$1) \quad z_i' + y_i' = rt_{i-1}' + w_i' \quad (25)$$
$$2) \quad w_i' + t_i' = rt_{i-1}'' + w_i'' \quad (26)$$
$$3) \quad s_i' = w_i'' + t_i''. \quad (27)$$

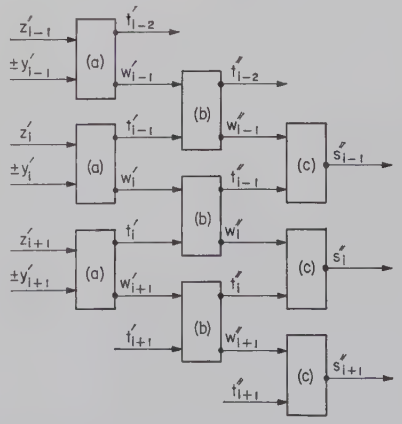


Fig. 2—Section of a two-transfer adder for modified signed-digit representation.

[10] J. E. Robertson, private communication.

The digits z_i', y_i' and s_i' are digits of a *modified* signed-digit representation.

For any radix $r \geq 2$, w_i' assumes $r+1$ values. For any *even* radix r_e, the values of w_i' range from $\frac{1}{2}r_e$ to $-\frac{1}{2}r_e$. In step 1) a transfer digit is generated whenever possible, *i.e.*, whenever $|z_i'+y_i'| \geq \frac{1}{2}r_e$, $|t_{i-1}'| = 1$; in step 2) a transfer digit is generated only if it is mandatory, *i.e.*, $|t_{i-1}''| = 1$ only if $|w_i'+t_i'| = \frac{1}{2}r_e+1$. Under these conditions t_i' and t_i'' cannot be both $+1$ or both -1, and s_i' assumes only r_e+1 values which are the same as the values of w_i'

$$\left\{-\tfrac{1}{2}r_e, \cdots, -1, 0, 1, \cdots, \tfrac{1}{2}r_e\right\}. \tag{28}$$

These r_e+1 values are the required values of one digit z_i' or y_i' for any even radix $r_e \geq 2$ if addition is performed according to (25)–(27). For instance, for the radix $r=2$ the required values are -1, 0, 1; for $r=4$ the required values range from -2 to 2; and for $r=10$ from -5 to 5.

For any *odd* radix $r_0 \geq 3$ the same procedure of addition applies if r_0+1 values of w_i' are chosen, ranging from $\pm\frac{1}{2}(r_0+1)$ to $\mp\frac{1}{2}(r_0-1)$. Only r_0+1 values of s_i' are required, which are the same as the values of w_i'. However, the principal advantage of two-transfer addition is lost because one more value of a digit must be introduced to provide the additive inverse of $\pm\frac{1}{2}(r_0+1)$, and r_0+2 values of a subtrahend digit y_i' are required at the input of the adder.

The potential advantages of two-transfer addition and of the resulting modified signed-digit representations are the use of radix $r=2$ (with digit values -1, 0, 1) and the decreased redundancy requirement (only $r+1$ values of a sum digit are required), especially for even radices. The disadvantages are the more complicated process of addition and the greater length of transfer digit propagation. It is interesting to observe that the procedure of two-transfer addition corresponds to the method of addition used for the "coincident carry-borrow storage" representation.[4]

Generally, it may be concluded that the lower limit on the required redundancy of one digit is a function of the number of digital positions over which a signal is allowed to propagate. If no redundancy exists and each sum digit assumes only r values, a sum digit s_i is the function of all the addend digits z_i and augend digits y_i to the right, *i.e.*, $s_i = f(z_i, y_i, z_{i+1}, \cdots, z_m, y_m)$. If each sum digit assumes $r+1$ values, we have $s_i = f(z_i, y_i, z_{i+1}, y_{i+1}, z_{i+2}, y_{i+2})$, and the operation is two-transfer addition. If each sum digit assumes $r+2$ values or more, we have $s_i = f(z_i, y_i, z_{i+1}, y_{i+1})$ with the restriction $r \geq 3$, and the addition is totally-parallel.

III. Signed-Digit Arithmetic

The preceding section presented the properties of a class of number representations which satisfy the four requirements 1) to 4) postulated in Section II-A. Given the class of signed-digit representations described by Definition 3, this section will describe the execution of addition, subtraction, shifting, multiplication and division for signed-digit numbers.

A. Elementary Arithmetical Operations

Addition, subtraction, left shifting and right shifting are the elementary operations for a signed-digit arithmetic unit. Multiplication and division are executed as sequences of additions or subtractions and shifts.

The addition of two signed-digit numbers is performed as the totally-parallel addition of all corresponding digits z_i and y_i according to (1) and (2). The block diagram of a signed-digit adder is shown in Fig. 1. To perform subtraction, first the sign of the subtrahend is changed by inverting the signs of all nonzero digits, then totally-parallel addition of corresponding digits is performed. Given the allowed values of w_i as the sequence: $w_{\min}, \cdots, -1, 0, 1, \cdots, w_{\max}$; the rules for forming w_i, t_{i-1} and s_i are as follows:

$$w_i = (z_i + y_i) - rt_{i-1}, \tag{29}$$

where

$$\begin{aligned} t_{i-1} &= 0 \text{ if } w_{\min} \leq z_i + y_i \leq w_{\max}, \\ t_{i-1} &= 1 \text{ if } z_i + y_i > w_{\max}, \\ t_{i-1} &= -1 \text{ if } z_i + y_i < w_{\min}; \\ s_i &= w_i + t_i. \end{aligned} \tag{30}$$

It is interesting to note that serial addition or subtraction of signed-digit numbers may be initiated with the most significant digits. Example 1 demonstrates the addition of two radix 10 signed-digit numbers.

Example 1—Signed-Digit Addition (Radix 10): The allowed digit values are: for w_i: 5, 4, 3, 2, 1, 0, $\bar{1}$, $\bar{2}$, $\bar{3}$, $\bar{4}$, $\bar{5}$;
for t_i: 1, 0, $\bar{1}$;
for s_i, z_i, y_i: 6, 5, 4, 3, 2, 1, 0, $\bar{1}$, $\bar{2}$, $\bar{3}$, $\bar{4}$, $\bar{5}$, $\bar{6}$.

Negative values are identified by a bar above the integer. The radix 10 signed-digit operands are:

augend z: $1.\bar{3}65\bar{1}\bar{4}$, algebraic value $Z = 0.76486$
addend y: $0.\bar{4}053\bar{1}$, algebraic value $Y = -0.39471$.

The procedure of addition is as follows:

augend z:	1.	$\bar{3}$	6	5	$\bar{1}$	$\bar{4}$
addend y:	0.	$\bar{4}$	0	5	3	$\bar{1}$
Step 1):	0+1.	$\bar{1}0$+3	10+$\bar{4}$	10+0	0+2	0+$\bar{5}$
Step 2):	$\bar{1}$	1	1	0	0	
sum s:	0.	4	$\bar{3}$	0	2	$\bar{5}$

The sum is s: $0.4\bar{3}02\bar{5}$, algebraic value $S = 0.37015$.

For the absolute range limits of (15), the limits for the sum or difference of two numbers $S = Z \pm Y$ are

$$|S| \leq 2 + 2\frac{r^{-1} - r^{-m}}{r-1} |z_i|_{\max}. \tag{31}$$

The overflow detection rules (16) and (17) assure that $|s_0| \leq 2$ will always hold and that $|t_{-1}| = 1$ may exist only for radix $r=3$, since $|w_0| = 2$ is allowed for all

radices $r>3$. The interim sum digits w_0, w_1 and transfer digits t_{-1}, t_0, t_1 contain all the necessary overflow information before s_0 and s_1 have been formed, and the overflow indication may be generated simultaneously with the sum digits. Immediate standardization may be performed in floating-point addition.

Shifting (left and right) of signed-digit numbers is performed in the conventional manner. If the digits z_i of a number pass through the adder logic during the shifting operation, transfer digits t_{i-1} will be generated when $z_i>w_{\max}$ or $z_i<w_{\min}$ and added to the interim sum w_{i-1} at the left according to (29) and (30). During the right shift of one position, $z_0=0$ is inserted as the most significant digit of the shifted number. If this shift includes transfer generation, the least significant digit z_m in the shifting register will retain the property $|z_m|\leq|z_i|_{\max}-1$, which is useful in multiple precision operations. Before the left shift of one position, the inspection of digits z_1 and z_2 will predict overflow after the shift. If the left shift includes transfer generation, digit z_3 must be inspected for a potential transfer digit generation.

B. Multiplication

Signed-digit multiplication is performed as a sequence of additions or subtractions and right shifts. Given the multiplicand z and the $m+1$ digits-long-multiplier y

$$\left(\text{algebraic value } Y=\sum_{i=0}^{m} y_i r^{-i}\right)$$

in radix $r\geq 3$ signed-digit representations, the product $p_{m+1}=zy+p_0r^{-m}$ is formed by the following recursive process:

$$p_{m+1}=p_m+zy_0 \tag{32}$$

$$p_{j+1}=\frac{1}{r}(p_j+zy_{m-j}) \quad \text{for } j=0,1,2,\cdots,m-1. \tag{33}$$

Here the p_{j+1} are partial products, p_0 is the initial number in the accumulator register and y_{m-j} is the multiplier digit sensed during the jth step of multiplication.

Temporary overflow may occur in the sum p_j+zy_{m-j} before the right shift which forms p_{j+1}. If p_j and z are in range, ($|p_j|\leq|z|_{\max}$, $|z|\leq|z|_{\max}$), we have

$$|p_j+zy_{m-j}|\leq|z|_{\max}(|y_{m-j}|_{\max}+1), \tag{34}$$

where $|z|_{\max}$ is the maximum allowed magnitude of a number. Since $|y_{m-j}|_{\max}\leq r-1$ always holds, p_{j+1} will also be in range. To store the temporary sum (before right shift) one more storage position (s_{-1}) must be allowed.

It is necessary to note that although both z and y do not indicate overflow, the product zy may indicate overflow if either z or y or both were in the potential overflow indication range given by (19). A test for overflow is necessary at the end of multiplication. If the temporary overflow storage position s_{-1} is available, the correct value of the product is retained in case of overflow.

Example 2 demonstrates signed-digit multiplication for radix 10 numbers.

Example 2—Signed-Digit Multiplication (Radix 10): The allowed digit values are given in Example 1. The radix 10 signed-digit operands are

multiplicand z: $0.\bar{4}62$, algebraic value $Z=-0.338$
multiplier y: $1.\bar{3}1\bar{5}$, algebraic value $Y=0.705$.

The procedure of multiplication is as follows:

step	digit y_{m-j}	quantity	representation	next adder operation
$j=0$	$y_3=\bar{5}$	p_0	0.000	
		zy_3	$2.\bar{3}\bar{1}0$	add
		p_0+zy_3	$2.\bar{3}\bar{1}0$	shift right
		p_1	$0.2\bar{3}\bar{1}0$	
$j=1$	$y_2=1$	zy_2	$0.\bar{4}62$	add
		p_1+zy_2	$0.\bar{2}\bar{3}10$	shift right
		p_2	$0.0\bar{2}\bar{3}10$	
$j=2$	$y_1=\bar{3}$	zy_1	1.014	add
		p_2+zy_1	$1.00\bar{3}\bar{1}0$	shift right
		p_3	$0.100\bar{3}\bar{1}0$	
$j=3$	$y_0=1$	zy_0	$0.\bar{4}62$	add
		p_4	$0.\bar{2}\bar{4}2\bar{3}\bar{1}0$	end

The product $zy=p_4$ is $0.\bar{2}\bar{4}2\bar{3}\bar{1}0$, algebraic value $ZY=-0.238290$.

It should be noted that the greatest absolute value of the multiplier digit y_{m-j} in (33) is $\frac{1}{2}r_e+1$ for even radices and $\frac{1}{2}(r_0+1)$ for odd radices when the multiplier y is in the minimal redundancy representation. By serial recoding of y (before sensing of y_{m-j}) to the minimal canonical (nonredundant) form, the greatest absolute value of y_{m-j} is reduced to $\frac{1}{2}r_e$ for even radices and $\frac{1}{2}(r_0-1)$ for odd radices. In this case, for the radix 3, the values of y_{m-j} are -1, 0, 1; for radix 4 they are -1, 0, 1 and 2 or -2; for radix 10 the ten required values are -4 to 4 and 5 or -5. Since y_{m-j} has values from 0 to $r-1$ in conventional representation, fewer additions (or fewer multiple-generating circuits) will be necessary on the average for one multiplication in signed-digit representation.

Eq. (24) shows that $n\leq\frac{1}{2}(r_0+1)$ or $n\leq\frac{1}{2}r_e$ digits can be simultaneously added for signed-digit numbers. If the multiplier y is recoded to the minimal canonical form, only one addition per step of (33) is sufficient for any odd radix $r_0\geq 3$ when each stage of the adder is designed to accept up to $\frac{1}{2}(r_0-1)$ inputs of the multiplicand digit z_i. The resulting product will be in the maximal redundancy form. For any even radix $r_e\geq 4$, two additions per step of (33) will be needed when $|y_{m-j}|=\frac{1}{2}r_e$. For all smaller magnitudes of y_{m-j} one addition per step is sufficient. This shows that the use of higher redundancy in the representation and of greater adder complexity permits very fast signed-digit multiplication.

C. Division

Signed-digit division is performed as a sequence of additions or subtractions and left shifts. The method of division which is most readily applicable to signed-digit representations is due to Robertson.[11] The representation of the quotient digits in this method may be redundant; however, the redundancy allows an inexact selection of quotient digits. In a signed-digit representation the sign of the dividend or a partial remainder is not readily available for inspection if several most significant digits are zero. The magnitude, however, may be estimated, with a limited uncertainty, from the inspection of a few most significant digits. The Robertson division method which requires such an estimate is therefore preferable to restoring or nonrestoring division, both of which require knowledge of the sign for exact selection of quotient digits.

Given the dividend z and the divisor d, the quotient digits q_j are generated by the following recursive process:

$$p_{j+1} = rp_j - dq_{j+1} \quad \text{for} \quad j = 0, 1, 2, \cdots, m-1 \tag{35}$$

$$p_0 = z - dq_0. \tag{36}$$

Here the p_j are partial remainders, p_m is the remainder and q is the quotient ($m+1$ digits long) with the algebraic value

$$Q = \sum_{i=0}^{m} q_i r^{-i}.$$

During each step of division the quotient digit q_{j+1} must be chosen which has a value such that the next partial remainder p_{j+1} in (35) is within the same allowed range as p_j. This range is a function of the magnitude of divisor d. If $|z|$ is not in this range, the choice of $|q_0| = 1$ in (36) must bring p_0 into allowed range; otherwise the quotient overflows. Every quotient digit q_{j+1} in (35) is assigned the value which satisfies the condition

$$|rp_j - dq_{j+1}| \leq c|d|, \tag{37}$$

where c is the *range test constant* whose allowed range is determined by the choice of the allowed values of the quotient digit. A *comparison circuit* must be constructed which performs (37); the value $c = \frac{1}{2}$ is the most practical choice within the allowed range.[12] If the representation of the quotient digits is redundant (q_{j+1} assumes more than r values), the comparison (37) may be inexact, that is, it is sufficient to perform the comparison between truncated values of $|rp_j - dq_{j+1}|$ and $|d|$. To facilitate the comparison (37) the divisor d must be standardized before division; otherwise, very great precision of comparison is required when $|d|$ is small.

The choice of allowed values for the quotient digits q_j is governed by two considerations. The least number of additions (or multiple-generating circuits) will be required for one division when the digits q_j assume the least possible number of values. However, at least minimal redundancy in representation of q_j is necessary to allow an inexact selection of quotient digits. In case of minimal redundancy the values of q_j range from $-\frac{1}{2}r_e$ to $\frac{1}{2}r_e$ (a total of $r+1$ values) for even radices $r_e \geq 4$, and from $-\frac{1}{2}(r_0+1)$ to $\frac{1}{2}(r_0+1)$ (a total of $r+2$ values) for odd radices $r_0 \geq 3$. For these values of q_j, it is sufficient to compare the first four digits of the test quantities in (37).

In the simplest mechanization of (35) and (36), the standardized divisor d is repetitively added to (if signs of d and rp_j disagree) or subtracted from (if signs of d and rp_j agree) the shifted partial remainder rp_j, until the condition

$$\left| \sum_{i=0}^{3} z_i r^{-i} \right| \leq \left| \frac{1}{2} \sum_{i=0}^{3} d_i r^{-i} \right| \tag{38}$$

is detected by the comparison circuit, indicating that p_{j+1} has been generated. Here d_i ($i=0, 1, 2, 3$) are the first four digits of d and z_i ($i=0, 1, 2, 3$) are the first four digits in the accumulator register, which contains the dividend z at the start of division, and later—the partial remainders and incomplete partial remainders. The value of q_{j+1} is equal to the number of additions (sign of q_{j+1} is minus) or the number of subtractions (sign of q_{j+1} is plus) required to generate p_{j+1}. If (38) is satisfied immediately after the left shift of p_j, $q_{j+1}=0$ is the required value. In any case, p_{j+1} is now shifted left and the procedure repeated to generate q_{j+2} and p_{j+2} until $m+1$ quotient digits have been generated. A numerical example of this process of division is given in Example 3.

The above-presented (binary) version of Robertson division requires the least amount of special circuitry. Quotient digits q_j may be generated during one addition cycle if more comparator circuits and circuits which generate multiples of the divisor are added to the arithmetic unit.[11]

Example 3—Signed-Digit Division (Radix 10): The allowed digit values are given in Example 1. The values of quotient digits q_j are 5 to $\bar{5}$ inclusive. The radix 10 signed-digit operands are

dividend z: $0.\bar{2}\bar{3}6\bar{4}15$, algebraic value $Z = -0.224385$
divisor d: $1.\bar{3}1\bar{5}$, algebraic value $D = 0.705$.

The test quantity is

$$\left| \frac{1}{2} \sum_{i=0}^{3} d_i r^{-i} \right|,$$

algebraic value $T = 0.3525$.

[11] J. E. Robertson, "A new class of digital division methods," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-7, pp. 218–222; September, 1958.

[12] Derivations of the range test constant, the precision requirement in comparison, and other results presented here appear in the Appendix.

The procedure of division is as follows:

step	quantity	representation	test	next adder operation	q_{j+1}
$j=-1$	z	$0.\overline{2}36\overline{4}15$	$0.224<T$	shift	$q_0=0$
$j=0$	rp_0	$\overline{2}.36\overline{4}150$	$2.244>T$	add d	
	d	$1.3\overline{1}5$			
	rp_0+d	$\overline{2}.5\overline{4}1$	$1.539>T$	add d	
	d	$1.3\overline{1}5$			
	rp_0+2d	$\overline{1}.2\overline{3}4$	$0.834>T$	add d	
	d	$1.3\overline{1}5$			
	p_1	$0.\overline{1}31$	$0.129<T$	shift	$q_1=\overline{3}$
$j=1$	rp_1	$\overline{1}.3\overline{1}1500$	$1.289>T$	add d	
	d	$1.3\overline{1}5$			
	rp_1+d	$\overline{1}.42\overline{4}$	$0.584>T$	add d	
	d	$1.3\overline{1}5$			
	p_2	0.121	$0.121<T$	shift	$q_2=\overline{2}$
$j=2$	rp_2	1.215000	$1.215>T$	add $-d$	
	$-d$	$\overline{1}.3\overline{1}5$			
	rp_2-d	$0.5\overline{1}0$	$0.510>T$	add $-d$	
	$-d$	$\overline{1}.3\overline{1}5$			
	p_3	$0.\overline{2}05000$	$0.195<T$	end	$q_3=2$

The quotient q is $0.\overline{3}\overline{2}2$, algebraic value $Q=-0.318$; the remainder p_3 is $0.\overline{2}05$, algebraic value $P_3=-0.195\times10^{-3}$.

D. Roundoff and Multiple Precision Operations

Addition and subtraction of numbers whose length is a multiple k of the $m+1$ digits long adder is performed as a sequence of k ordinary additions or subtractions in signed-digit arithmetic. The least significant parts are brought up first, and if a transfer digit t_{-1} is generated out of the most significant adder stage, it is applied as an input transfer digit t_m during the following operation on the two next higher-significance parts of the operands. This is always possible because $t_m=0$ for any two single-length operands. It should, furthermore, be noted that a single-length number may be added to or subtracted from the more significant half of a double-length number without affecting the less significant half in any manner. This is due to the general property of signed-digit representations that every digit contains its own sign information and therefore any section of a representation is a complete signed-digit representation in its own right.

The operation of roundoff is often required to reduce a number to the standard length. Roundoff for signed-digit numbers is performed by truncation. If the allowed values of a digit z_i are chosen symmetrically around zero $(-a, \cdots, -1, 0, 1, \cdots, a)$ and we assume that every one of these values will occur with the same probability, then the average error which is introduced by truncation is zero and the roundoff is without bias.

The knowledge of the number of significant digits in a fixed-length representation may be desirable in digital computer arithmetic, especially in floating-point operations. Because roundoff is performed by truncation for signed-digit numbers, it is possible to maintain a continuous indication of nonsignificant digits during addition, subtraction and left shifts. A new digit value $z_i=0'$ called the *space-zero* (distinct from the *value-zero* $z_i=0$) may be introduced to identify nonsignificant positions in a signed-digit representation. The following rule applies to the space-zero value $0'$:

$$z_i \pm 0' = 0' \text{ for all values of } z_i. \tag{39}$$

In the first step of totally-parallel digit addition $(z_i \pm 0' = rt_{i-1}+w_i)$, $t_{i-1}=0$ and $w_i=0'$ are formed; the second step yields the sum digit $s_i=w_i+t_i=0'+t_i=0'$. Space-zero value digits may appear only as a continuous string of digits extending to the left from the least significant position of a register.

The nonsignificant positions in input numbers are represented by space-zero digits. During left shifts and during standardization in floating-point arithmetic space-zero digits are placed in the least significant position of the shifting register. When two signed-digit numbers are added or subtracted, the space-zero digits in the number with fewer significant digits will round off the sum or difference to the lesser precision. The space-zero digits will maintain a continuous indication of precision for all signed-digit numbers in the machine.

IV. Logical Design of Signed-Digit Adders

One problem of practical interest is the logical design of an adder to perform totally-parallel addition and subtraction with numbers in signed-digit representation. A parallel adder for signed-digit numbers of m-digit length consists of m identical *digit-adders* (also stage-adders[7]) which are linked to their immediate neighbors by transfer digit lines, an output to the left and an input at the right. Fig. 1 shows a block diagram with the required interconnections, inputs and outputs of digit-adders. All transfer digits t_i and interim sum digits w_i are formed simultaneously, thus the addition time for numbers of any length is equal to the addition time for one digit-adder. The objective of a digit-adder is to execute the operations described by (29) and (30) for digits of a given signed-digit representation, which is defined by the choice of the radix r and the allowed values of digits z_i, y_i, w_i and t_i. The allowed ranges of choice have been derived in the preceding sections. Minimal-redundancy representations require the least storage capacity for the values of a digit and therefore are preferable to representations with higher redundancy. Furthermore, less complicated digit-adder logic may be expected when the least possible number of digit values is employed.

The choice of the radix r depends on the desired balance between the increase in storage requirements and the logical complexity of one digit-adder. The relative increase in storage capacity requirements diminishes when r is large; however, one digit-adder must accept more values of a digit and the logical circuits become

more complex. Few digit values are also preferable in multiplication and division. For instance, radix 4 requires seven values (-3 to 3) for the digits of a subtrahend at the input to the digit-adder, at least six of these values (-3 or 3 may be avoided) for the sum digit s_i, and in storage, five values of the quotient digits (-2 to 2) and at least four of these values (-2 or 2 may be avoided) for the multiplier digits. Three binary storage elements per digit are required, compared to two for conventional radix 4 representations. Three binary storage elements per digit are sufficient for all radices $3 \leq r \leq 6$, four elements for radices $7 \leq r \leq 14$. The radix 10 (with 13 values) requires four elements per digit, the same as for conventional representations; the required values of quotient digits are -5 to 5. The space-zero value and additional redundancy may be introduced for radix 10 and many other radices without increasing the storage requirements.

The choice of the binary representation in storage of the allowed values for one digit affects the complexity of logical circuits which perform (29) and (30). A one digit-adder may be treated as a multiple-input and multiple-output switching circuit which is specified by these rules of addition. Alternatively, a digit-adder may be designed as a small conventional adder which consists of basic building blocks (such as half-adders) and has appropriate modifications to generate t_{i-1} and w_i according to the specifications of (29) and (30). Digit values are represented by binary digits corresponding to the storage elements, and conventional methods are used to represent negative values. The minimization of the logic for a digit-adder as a function of the specific storage representation of digit values, of the redundancy of digit representation, and of the choice of basic logic circuits remains as an interesting unsolved program for further investigation.

As an example, logical design has been performed for a radix 4 digit-adder.[13] The "building-block" approach was used in this design and conventional adder design techniques were employed. The seven values of a radix 4 digit (-3, -2, -1, 0, 1, 2, 3), were represented by three binary digits weighted -4, 2, and 1. The radix 4 digit-adder required the equivalent logical circuitry of approximately 12 half-adders of the conventional type; this is 2.5 to 3 times more than required by one position of a conventional radix 4 adder with serial carry propagation. The comparison is more favorable when the auxiliary circuitry of the conventional adders (carry acceleration, carry-completion sensing, conditional corrections) is considered. An investigation of a radix 10 digit-adder showed that it was about two times more complex than one stage of a conventional excess-three code adder. The experimental designs were intended to serve as an existence demonstration and an indication of the order of complexity of a digit-adder. No claim is made for minimal complexity of these designs; furthermore, design problems which are not directly affected by the totally-parallel addition requirements, such as gating, synchronization or timing, and arithmetic control were not considered.

[13] Avizienis, *op. cit.*, pp. 63–70.

V. Conclusion

A class of redundant number representations which permit the elimination of carry-propagation chains in addition and subtraction has been demonstrated. Unique representation exists for the zero value of a number, and all sign information is contained by the individual digits. No special sign digit is required; this eliminates preliminary and terminal corrections caused by sign digits.

Procedures for arithmetical operations have been described for the class of signed-digit representations. The time required for addition or subtraction is independent of the length of the operands; also, simultaneous addition of several numbers is possible. Convenient procedures exist for overflow detection, roundoff, and multiple-precision operations. Multiplier and quotient digits assume values which require fewer adder operations than an equivalent multiplication or division in conventional representation. A number in signed-digit representation requires more allowed values and more storage capacity per digit than the same number in conventional representation. The logical circuits are more complex for a signed-digit adder. The minimization of digit-adders remains a problem for further investigation.

A signed-digit number of arbitrary length consists of a positionally-ordered collection of complete one-digit long representations; this property leads to useful simplifications in arithmetical operations. Partition of a signed-digit number at any position yields two complete signed-digit numbers whose sum is equal to the original number. In a conventional parallel adder all iterative adder circuits are bound together by the requirement for carry propagation. In a signed-digit parallel adder, each iterative digit-adder depends only on its own inputs and the information generated by its immediate neighbor to complete one cycle of addition. It may be concluded that signed-digit representations offer the means for a completely parallel execution of arithmetical operations in a digital computer.

Appendix

The Division Comparison

The maximum allowed magnitude of a partial remainder p_j in (35) and (36) is a function of $|d|$, *i.e.*, $|p_j|_{\max} = k|d|$.

Two requirements are to be satisfied:

Requirement 1: The greatest partial remainder should be returnable into range after a shift by means of the greatest quotient digit $|q_j|_{\max}$

$$r k |d| - |d| \, |q_j|_{\max} \leq k |d|. \tag{40}$$

Eq. (40) yields the upper bound of k as

$$k \leq \frac{|q_j|_{\max}}{r-1}.$$

Requirement 2: If the shifted partial remainder is outside the range ($|rp_j| = k|d| + \delta > k|d|$) by the least detectable difference δ, a single reduction of rp_j by d should yield a partial remainder within range, *i.e.*,

$$k|d| + \delta - |d| \geq -k|d|. \tag{41}$$

Eq. (41) yields the lower limit of k as

$$k \geq \frac{1}{2} - \frac{1}{2}\frac{\delta}{|d|} \approx \frac{1}{2},$$

where the approximation $k=\frac{1}{2}$ is satisfactory since $|d|_{\max} \geq 1$ always holds.

The division requirements are satisfied for any value of k in the range limited by the above requirements, that is, for

$$\frac{1}{2} \leq k \leq \frac{|q_j|_{\max}}{r-1}. \tag{42}$$

The value of k may be allowed to vary within the range of (42); therefore, an *inexact* comparison is possible in (37). The allowed error of truncation must be determined. The quantities which are compared in (37) are the divisor $d=d'+d''$ and the quantity $p=p'+p''$. Here p is first the dividend z, then a shifted partial remainder rp_j or a diminished shifted partial remainder.

The test is to detect the condition $|p'| \leq c|d'|^*$ where c is the *range test constant.* An uncertainty of comparison occurs because of the uncompared lower-significance parts d'' and p''. When both p and d are truncated behind the position $i=h$, we denote $|p''|_{\max} = |d''|_{\max} = \Delta$, and determine the maximum value of Δ which satisfies $|p_j|_{\max} = k|d|$. In the test $|p'| \leq c|d'|^*$ the precision of $c|d'|^*$ is the same as of $|p'|$, that is, $h+1$ digits. Therefore $c|d'| = c|d'|^* + e$, where $0 \leq e < r^{-h}$.

Two conditions must be satisfied:

Condition 1: When $|p'| \leq c|d'|^*$ is indicated, $|p| = k_{\max}|d|$ must hold under the worst conditions, which occur when $|p| = |p'| + \Delta$, $|d| = |d'| - \Delta$ and $|p'| = c|d'|^*$ with $e=0$. This yields

$$\Delta_1 = |d| \frac{k_{\max} - c}{1+c} \tag{43}$$

as the maximum allowable error of truncation.

Condition 2: When $|p'| > c|d'|^*$ is indicated, one more reduction of $|p|$ by $|d|$ should retain $|p| - |d|$ in range, that is, $|p| - |d| = -k_{\max}|d|$ must hold under the worst conditions which occur when $|p| = |p'| - \Delta$, $|d| = |d'| + \Delta$ and $|p'| = c|d'|^* + r^{-h}$ with $e = e_{\max}$. This yields

$$\Delta = |d| \frac{k_{\max} + c - 1}{1+c} + \frac{r^{-h} - e_{\max}}{1+c}, \tag{44}$$

and since $(r^{-h} - e_{\max})$ is a small positive quantity, a good approximation is

$$\Delta_2 = |d| \frac{k_{\max} + c - 1}{1+c}. \tag{45}$$

From the above results, the condition $\Delta \geq 0$ applied to (43) and (45) yields the range

$$1 - k_{\max} \leq c \leq k_{\max}, \tag{46}$$

in which no truncation uncertainty is allowed ($\Delta = 0$) for $c = k_{\max}$ or $c = 1 - k_{\max}$, and an uncertainty given by (43) and (45) may exist for $1 - k_{\max} < c < k_{\max}$, where

$$k_{\max} = \frac{|q_j|_{\max}}{r-1} \text{ from (42).}$$

Setting $\Delta_1 = \Delta_2$, we get $c = \frac{1}{2}$ and

$$\Delta_{\max} = \tfrac{1}{3}(2k_{\max} - 1)|d| \tag{47}$$

as the greatest value of Δ as a function of c, since Δ_1 decreases and Δ_2 increases as c ranges from $1 - k_{\max}$ to $k_{\max}$.

When the divisor d is standardized, its least value is

$$|d|_{\min} = \frac{1}{r} + r^{-2}\left(1 - \frac{|z_i|_{\max}}{r-1}\right) + r^{-m}\frac{|z_i|_{\max}}{r-1} \tag{48}$$

for $r \geq 3$, the range of (15) and overflow rules (16) and (17). Condition (48) occurs when $d_0 = 0$, $d_1 = d_2 = 1$ and $d_i = -|z_i|_{\max}$ for $i = 3, 4, \cdots, m$. Therefore $|d|_{\min} \geq 1/r$ holds for all signed-digit representations, and for standardized divisors we have

$$\Delta_{\max} = \frac{1}{3r}\left(2\frac{|q_j|_{\max}}{r-1} - 1\right). \tag{49}$$

This equation indicates the required precision of comparison for any choice of $|q_j|_{\max}$. For example, if $|q_j|_{\max} = r-1$ (maximum redundancy case), we have $\Delta_{\max} = 1/3r$, and it is sufficient to compare positions $i = 0, 1, 2$. For minimal redundancy and even radices

$$r_e \geq 4, \quad |q_j|_{\max} = \tfrac{1}{2}r_e \quad \text{and} \quad \Delta_{\max} = \frac{1}{3r_e(r_e-1)};$$

here the positions $i = 0, 1, 2, 3$ must be compared. For minimal redundancy and odd radices

$$r_0 \geq 3, \quad |q_j|_{\max} = \tfrac{1}{2}(r_0+1) \quad \text{and} \quad \Delta_{\max} = \frac{2}{3r_0(r_0-1)};$$

here the positions $i = 0, 1, 2, 3$ must be compared except for $r = 3$, where $i = 0, 1, 2$ are sufficient.

Derivation of Allowed and Required Digit Values

The conditions (1)–(4) establish the restrictions

$$2|z_i|_{\max} \leq r|t_i|_{\max} + |w_i|_{\max} \tag{50}$$

$$|t_i|_{\max} + |w_i|_{\max} \leq |z_i|_{\max} \tag{51}$$

on the maximum absolute values of t_i, w_i and z_i (or s_i and y_i). Given $|z_i|_{\max}=r-1$, we have the limiting conditions

$$2(r-1) = r|t_i|_{\max} + |w_i|_{\max} \quad (52)$$

$$r-1 = |t_i|_{\max} + |w_i|_{\max} \quad (53)$$

which are simultaneously satisfied by $|t_i|_{\max}=1$ and $|w_i|_{\max}=r-2$, leading to (7)–(9).

In the conversion procedure described by (5) and (6), the $2r-1$ values of x_i (from $-r+1$ to $r-1$) are to be recoded as $x_i=rt_{i-1}+w_i$. The values $x_i=-1$, 0, 1 require corresponding values $w_i=-1$, 0, 1, because (8) does not allow their recoding with $|t_i|>0$. At least r values of w_i are needed to recode all $2r-1$ values of x_i. In this case, $w_i=a$ is generated for each pair of values $x_i=a$ and $x_i=-(r-a)$, except for $w_i=0$ which is generated for $x_i=0$ only. Furthermore, in this case $w_{\max}+1$ is recoded as $r+w_{\min}$, where $w_{\max}$ is the greatest allowed value and $w_{\min}$ is the least allowed value of w_i, giving

$$w_{\max} - w_{\min} = r - 1, \quad (54)$$

which is the limiting case of (10). The value $|w_i|_{\max}$ is least when we choose $w_{\max}=|w_i|_{\max}=\frac{1}{2}(r_0-1)$ for odd radices r_0 and $w_{\max}=|w_i|_{\max}=\frac{1}{2}r_e$ for even radices r_e.

In simultaneous totally-parallel addition of n digits, the restrictions are

$$|z_i|_{\max} = |w_i|_{\max} + |t_i|_{\max} \quad (55)$$

$$n|z_i|_{\max} \leq r|t_i|_{\max} + |w_i|_{\max}, \quad (56)$$

which give the lower limit of $|t_i|_{\max}$ as

$$|t_i|_{\max} \geq \frac{n-1}{r-1}|z_i|_{\max}. \quad (57)$$

The upper limit of $|t_i|_{\max}$ is given by (55) as

$$|t_i|_{\max} \leq |z_i|_{\max} - |w_i|_{\max}. \quad (58)$$

The greatest range of $|t_i|_{\max}$ is obtained when the greatest allowed value of $|z_i|_{\max}=r-1$ and the least allowed value of $|w_i|_{\max}=\frac{1}{2}r_e$ or $|w_i|_{\max}=\frac{1}{2}(r_0-1)$ is used in (22). These choices also lead to the maximum values of n given in (24).

Acknowledgment

The author wishes to thank Prof. J. E. Robertson of the Digital Computer Laboratory, University of Illinois, Urbana, Ill., for his comments on this paper and for guidance and critical advice during the writing of the doctoral dissertation[7] on which this paper is based.

Computing Machine Aids to a Development Project*

C. W. ROSENTHAL†, MEMBER, IRE

Summary—**A system o integrated computer programs which provide useful machine assistance to the development of a digital system will be described. The individual programs are capable of such separate engineering tasks as verifying design data, optimally locating electronic logic packages on a chassis, routing interconnecting wires and preparing documents like wiring diagrams and wire running lists.**

The engineer using the system need not be a computer programmer, because the machine aids programs are called into operation with a simple mnemonic language keyed to the engineer's traditional tasks. This aspect of the work, as well as the accomplishment of the individual engineering tasks, is emphasized in this paper.

Introduction

AUTOMATION is generally understood to be a substitution of machine controllers for human controllers in industrial or commercial processes and procedures. Frequently the controllers are electronic computers actuating other machines or aiding human beings in their work. The machine controllers are introduced because they provide some combination of improvements in the economy, dependability, productivity, accuracy, or adaptivity to environment of a system.

A group of technical people developing a new system themselves form a system requiring the fulfillment of the roles of designers, draftsmen, manufacturers and operators. These people cooperate to originate, transcribe, distribute and interpret information. Now they, too, may work with the aid of computers. This latest type of automation has been called "design automation"[1] and for our purposes may be described as the use of an integrated set of computer programs to perform a sequence

* Received by the PGEC, August 17, 1960.
† Bell Telephone Labs., Inc., Murray Hill, N. J.

[1] M. Kloomok, P. W. Case, and H. H. Graff, "The recording, checking and printing of logic diagrams," *Proc. Eastern Joint Computer Conf.*, pp. 108–118; December, 1958.

of engineering tasks in a development project. These tasks include verification of design data, location of electronic logic packages, routing of interconnecting wires and the preparation of documents like wiring diagrams and wire running lists. The present paper describes an integrated set of IBM 704 programs—The Development Information Processing System (DIPS)—intended to perform all of the above and also to maintain magnetic tape records capable of ready alteration due to design changes. In describing the DIPS we will view it as an engineer-user might and then afterwards explain the way in which the constituent parts are accomplished.

DIPS users are provided with a mnemonic language that enables them to instruct the program to perform the selected operations by using a set of order words that can readily be identified with engineering tasks. DIPS interprets these order words and the computer carries out the prescribed tasks. The DIPS user need not know how to program the IBM 704 computer. He need only know the two dozen words in the DIPS vocabulary, the simple punctuation requirements for separating order words, and the operating procedures involved. Table I

TABLE I

DIPS Orders	
ADWIRE	NEW
ALTER, N	OLD, N
CHANGE	PAUSE
CONVRT	PLACE
COPY, N	PLCLST
DGRAM	PRELST
DRAWER	SAVE
ENDJOB	TITLE
FIXLST	TRA, Y
HAND	WEED
HIDE, A, B	WIRE
LOAD, N	WIRLST

contains the list of DIPS order words provided. Though not all the words will be defined here, a few will be described to aid in the understanding of the system's intent.

WIRE produces a routing of the wires needed to interconnect electronic logic packages. The route is specified by its configuration and the length of its segments. ALTERN provides for the alteration of the record of a wire as prescribed by a following data card. NEW calls in a program which introduces and checks new data on the following data cards. SAVE produces a magnetic tape file of the data for the equipment unit whose data is then in the computer's high speed store. WEED will erase the records identified on the data cards that follow and squeeze the remaining records together. This permits the selected erasure of old unneeded records or errors.

A flow chart of the DIPS procedure is shown in Fig. 1. The original data is obtained from a logic diagram for the equipment unit to be considered. In our application the basic unit is a chassis or, as we call it, a drawer whose equipment arrangement will be described below. The data on the logic diagram is tabulated and key punched by clerical personnel. One punched data card is produced per interconnecting wire. The DIPS programs are held on a library tape accessible to the computer. The order words instructing DIPS and selected by the engineer are contained in a punched card deck of about one-half dozen cards. The information on the data deck is processed by the library tape programs selected by the order words on the instruction deck to produce a new data tape and the specified documents. The new tape may be held and made accessible to the computer at a later time as changes in the equipment unit develop. These changes are introduced as one data card per wire change. It should be emphasized that the entire data deck need not be reintroduced when changes are made.

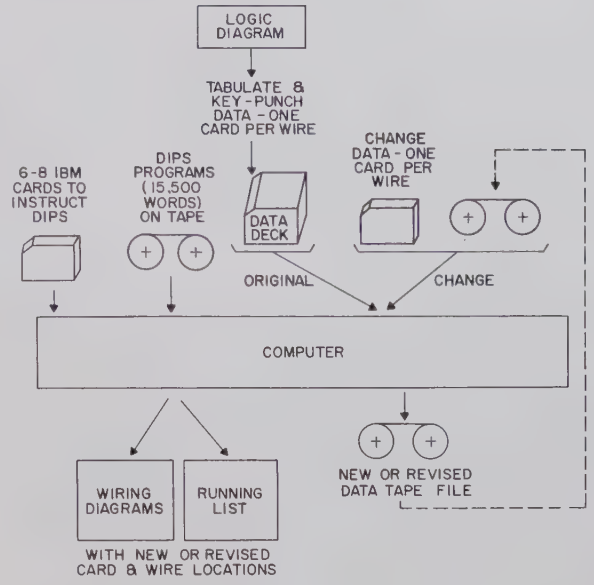


Fig. 1—DIPS procedure.

Fig. 2 shows representative data and order cards of the DIPS. The meaning of the information on the data card will become clearer when we describe the type of information introduced.

The DIPS processes information for one drawer at a time. However, programs exist, which are not part of DIPS, for producing such inter-drawer data as wire routing and wire type categorization. These programs are described in a publication by Morzenti of the Bell Telephone Laboratories.[2]

Information on the logic diagram (see Fig. 3) is obtained for DIPS processing as a result of the following sequence: 1) The logic designers prepare a hand drawn sketch [Fig. 3(a)] indicating the interconnection of specified logic nodes and labeling of certain wires. 2) To accommodate the packaging arrangement employed, the

[2] O. J. Morzenti, "Implications of Machine Aids to Design," AIEE Paper No. 61-104; February, 1961.

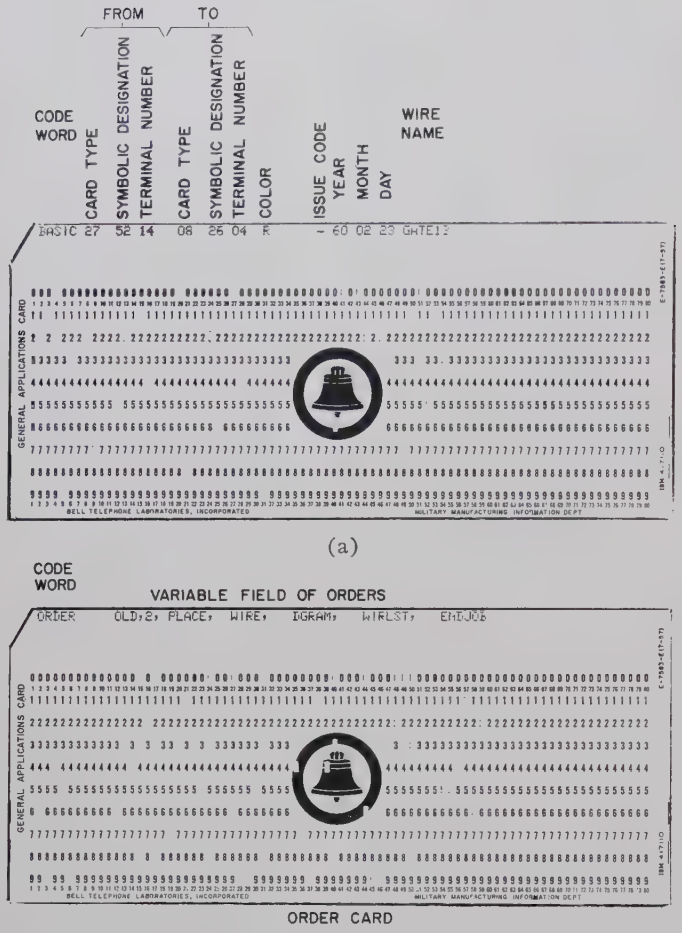
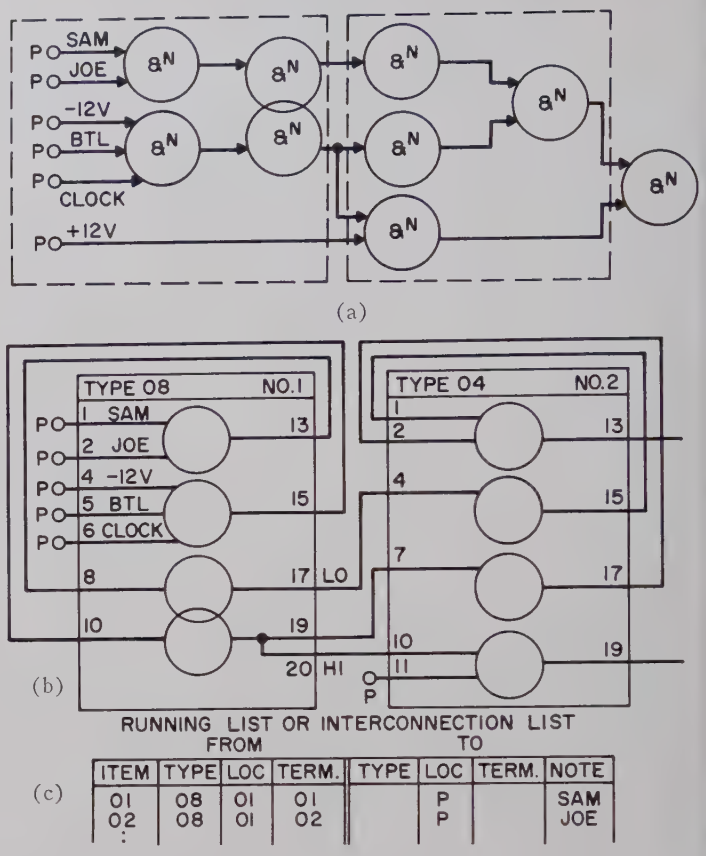


Fig. 2—DIPS cards. (a) Data. (b) Order.



ITEM	TYPE	LOC	TERM.	TYPE	LOC	TERM.	NOTE
01	08	01	01		P		SAM
02	08	01	02		P		JOE
:							

Fig. 3—Logic diagrams and resultant running list.

logic nodes must be implemented as, say, transistors on permissible packages. We shall call these packages cards, of which there are a small set of allowable card types. The engineer will select the particular card types on which the logic nodes are to appear and specify the terminals to be used. 3) He will also arbitrarily designate a symbolic location for these cards. The symbolic location is intended to distinguish among the cards utilized. The arbitrary symbolic location is shown in the upper right of a card, as represented in Fig. 3(b). The card type is indicated in the upper left. 4) Terminals are, and wire names may be, indicated alongside the wires. The symbol P indicates that the wire comes from another drawer and a unique connector terminal must be assigned to each P. 5) The redrawn diagram [Fig. 3(b)] is used to prepare a running list or interconnection list as shown in part as Fig. 3(c). There is one entry on the list per wire and the wire is identified by its start (FROM) and its end (TO) descriptions, by notes for wire name and by a unique item number. Running list data is then processed for computer input as one punched card per item. As an example of the magnitude of this task, a drawer containing 54 cards required 718 interconnecting wires and hence 718 punched cards.

The logic diagram from which the DIPS input data is obtained has a considerable amount of location information. In particular, logic nodes have been implemented by transistors on cards selected by an engineer. The engineer has also specified the terminals to be used. Computer selection of terminals, transistors and cards would be desirable in any extension of DIPS.

The DIP System may be ordered to perform its operations in any one of the sequences obtained by tracing paths in Fig. 4. The usual one will be followed below. Initial data is operated upon by the routine NEW to introduce and to check the new data and produce lists of detected errors, if any. Then, if indicated by the engineer's order words, the system would save the data as a tape file (SAVE), prepare a list of this preliminary information (PRELST) and/or go on to the routine called PLACE. The PLACE routine optimally locates the cards on a drawer and specifies the connector terminals for wires entering or leaving the drawer. At this point the computer run could conclude or go on. If the run concluded with the saved tape file, we would return another time by using the order OLD,N to retrieve file N. If changes are introduced, the order CHANGE is employed to revise the existing data. From this point the run may continue with the choices shown; *i.e.*, WIRE, ADWIRE, and/or PLCLST and then possibly more changes and beyond this various lists, a wiring diagram and a file of processed data.

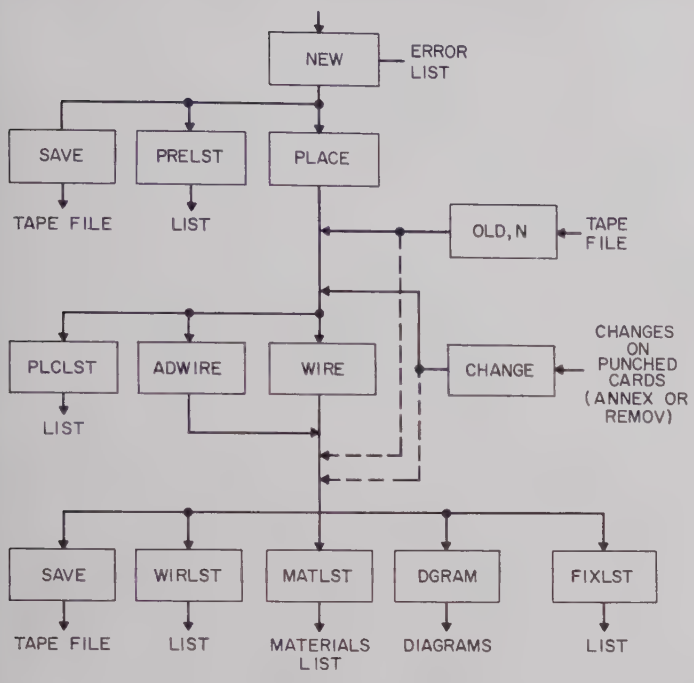


Fig. 4—Inter-relation of DIPS orders.

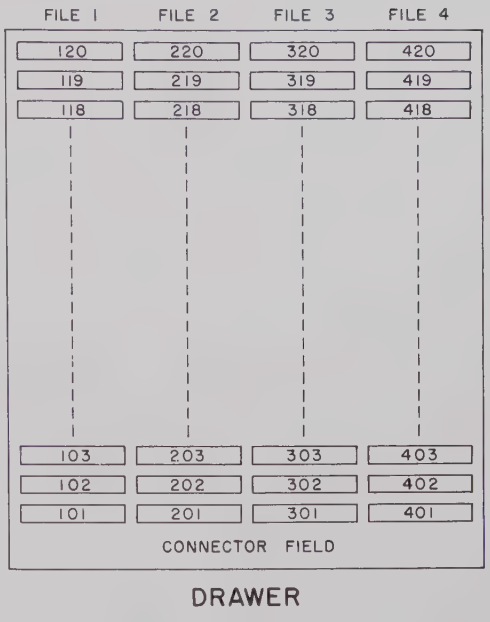


Fig. 5—Card arrangement on drawer.

Equipment Design Employed

The layout of mechanical hardware, upon which the routines of DIPS are based, must be considered before examining some of the DIPS programs for the individual orders. Fig. 5 indicates the manner in which cards are located and labeled on a drawer. They appear as four files of 20 cards each. A connector field for wires entering and leaving the drawer is also shown. This connector field contains one end of the wires originally designated as P on the input data. Though originally based on this equipment, configuration DIPS is now being modified for a somewhat different configuration.

The Optimum Location of Cards

It is the task of the program PLACE to locate the specified cards in such positions that will obtain a "practical" minimum on the total length of interconnecting wires. For this purpose, the PLACE routine utilizes a method proposed by Glaser.[3] This method involves the following sequence of steps:

1) The cards are initially positioned on the drawer in a randomly selected manner and the total length of interconnecting wire is calculated.
2) The card to which the largest number of wires are connected is selected.
3) This card is transposed with all other cards on the drawer and left in that transposition which yields the largest reduction in total wire length below the original total wire length.
4) The card with the next largest number of wires connecting to it is selected.
5) This card is transposed with all others and left in that transposition which produces the largest decrease in total wire length below that obtained in step 3).
6) This technique is employed for the remaining cards taken in decreasing order of number of connecting wires.
7) After the attempted transposition of the last card in the ordered list, the procedure starts over again with the card highest on the list. The starting position is the result of the preceding transposition.

Convergence of the above method of solution has turned out to be very rapid. Generally four passes through the list are sufficient to produce a situation in which succeeding passes improve upon the original total wire length by less than 1 per cent. A 1 per cent improvement test is built into the program to stop further passes. The time for a pass appears to be 3 to 5 minutes when using about 60 cards. When compared against the results of an engineer placing cards we have obtained an average reduction of 25 per cent in total wire length and many hours in elapsed time.

The Listing Programs

One program in the DIPS system, with several entry points, is used to prepare several types of wire running lists. There are four such lists, PRELST, PLCLST, FIXLST, and WIRLST. They are employed to depict information which is in various stages of processing. This enables the user to observe the system's results and introduce changes if he is not satisfied.

A portion of the results of WIRLST are indicated in Fig. 6. Titling information is added to the wire data

[3] R. H. Glaser, "A Quasi-Simplex Method for Designing Suboptimum Packages of Electronic Building Blocks," unpublished paper.

DATE. 30 MAY 60 SHEET 2 OF 18

COMMUNICATION SYSTEM

RUNNING LIST (WIRED)

CIRCUIT TITLE ZDC INTERFACE SUPV CKT
LOGIC DIAGRAM G-497130
REV.-

ORIGINAL DATE 30 MAY 60
ORD. ASSEM. G-497128
DRAWER NUMBER GS XXXXX

ITEM	FROM CARD TYPE	FROM JACK LOCATION	FROM TERMINAL NUMBER	TO CARD TYPE	TO JACK LOCATION	TO TERMINAL NUMBER	REMARKS		WIRE PATH H	WIRE PATH V
41	02	405	14A	02	405	23A		⌟	1.4	0.2
42	02	405	16B	30	406	21A		⌜	0.8	0.8
43	02	405	17A	02	405	GNDA		⌞	2.8	0.2
44	02	405	17B	02	405	GNDB		⌐	2.0	0.
45	02	405	20B	30	406	22A		⌟	0.3	0.7
46	02	405	22B	30	406	23A		⌜	0.2	0.8
47	02	405	23B	08	204	12B		⌞	11.7	1.1
48	02	405	24B	30	406	21B		⌝	0.5	1.1
49	02	405	25B	30	406	22B		⌞	0.5	0.8
50	02	405	26A	09	402	03B		⌝	3.5	2.4
51	02	405	27A	09	402	02B		⌞	3.8	2.7
52	02	405	28A	11	404	28B	+12V	1	0.	0.7
53	09	402	01A	09	401	01B	-12V	1	0.	0.7
54	09	402	17A	09	402	GNDA		⌝	2.8	0.2
55	09	402	17B	09	402	GNDB		⌐	2.0	0.
56	09	402	23A			P3-057	CAL-B	⌝	2.2	2.8
57	09	402	24A			P1-058	CAL-A	⌞	2.1	3.4
58	09	202	01A	09	201	01B	-12V	1	0.	0.7
59	09	202	23A			P3-024	SMG	⌞	1.8	2.8
60	09	202	24A			P2-026	AKN	⌝	1.4	3.0
61	09	202	17A	09	202	GNDA		⌞	2.8	0.2
62	09	202	17B	09	202	GNDB		⌐	2.0	0.
63	09	202	02B	01	105	23A		⌞	1.9	2.7
64	09	202	03B	01	105	25A		⌝	1.7	2.6
65	09	202	27A			P1-029	MBV-A	⌞	1.1	3.6
66	09	202	06B	02	205	24A		⌜	2.7	2.6
67	11	103	01A	09	101	01B	-12V	1	0.	1.6
68	11	103	02A	38	102	26B		⌞	3.6	0.7
69	11	103	05B	30	107	04A		⌞	0.2	3.4
70	11	103	08B	30	107	07A		⌝	0.2	3.7
71	11	103	11A	11	103	12A		⌝	0.2	0.2
72	11	103	12B	11	103	13B		⌜	0.2	0.2
73	11	103	14A	11	103	15A		⌟	0.2	0.2
74	11	103	15B	11	103	16B		⌞	0.2	0.2
75	11	103	17A	11	103	GNDA		⌞	2.8	0.2
76	11	103	17B	11	103	GNDB		⌐	2.0	0.
77	11	103	18A	11	103	19A		⌟	0.2	0.2
78	11	103	19B	11	103	20B		⌞	0.2	0.2
79	11	103	22A	11	103	23A		⌟	0.2	0.2
80	11	103	23B	30	107	19A		⌝	0.6	3.5

Fig. 6—Sample wire running list prepared by DIPS.

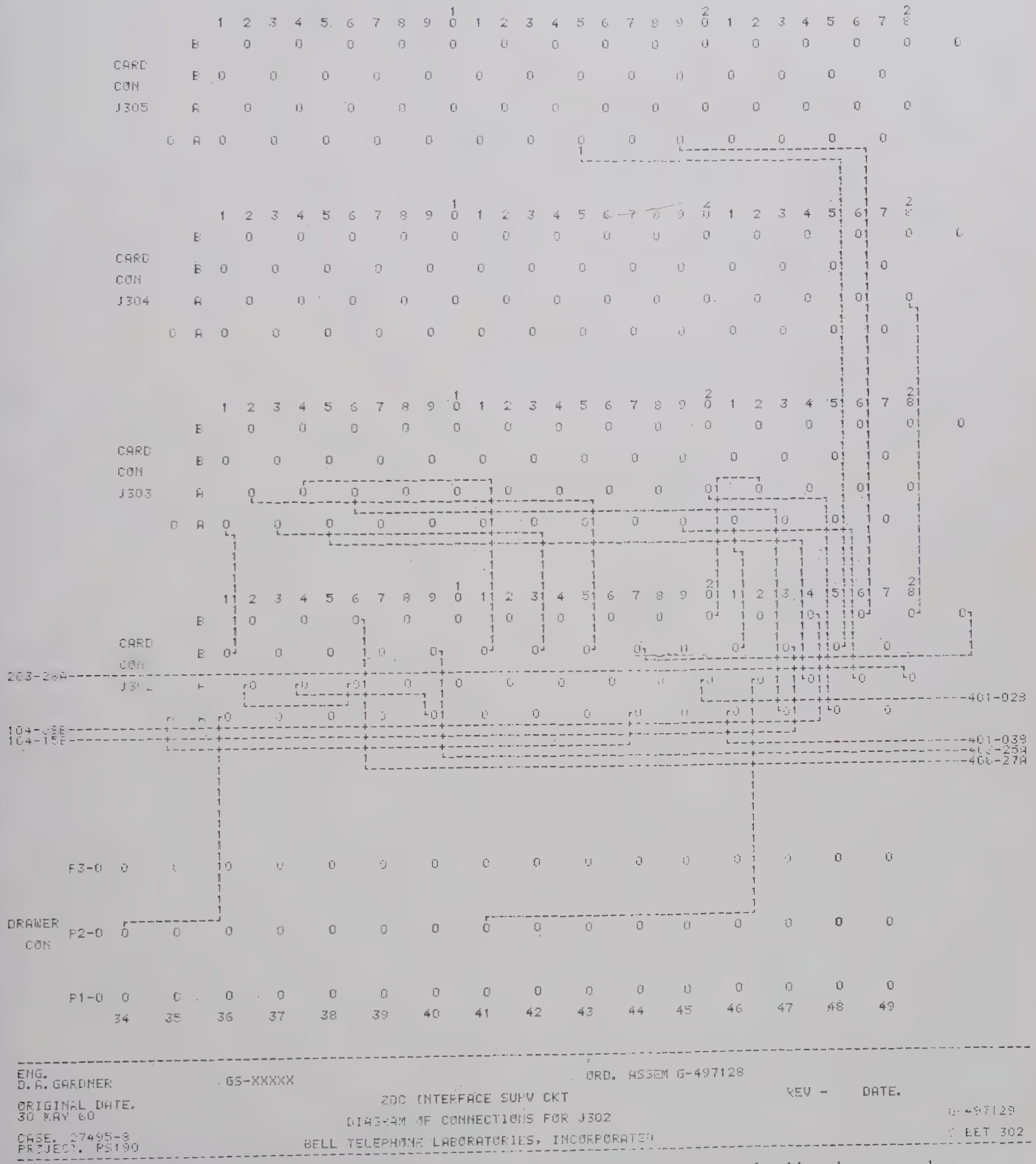


Fig. 7—Sample wiring diagram prepared by DIPS. Connectors above J305 are not used and have been cropped from the figure, although they appear in the computer output.

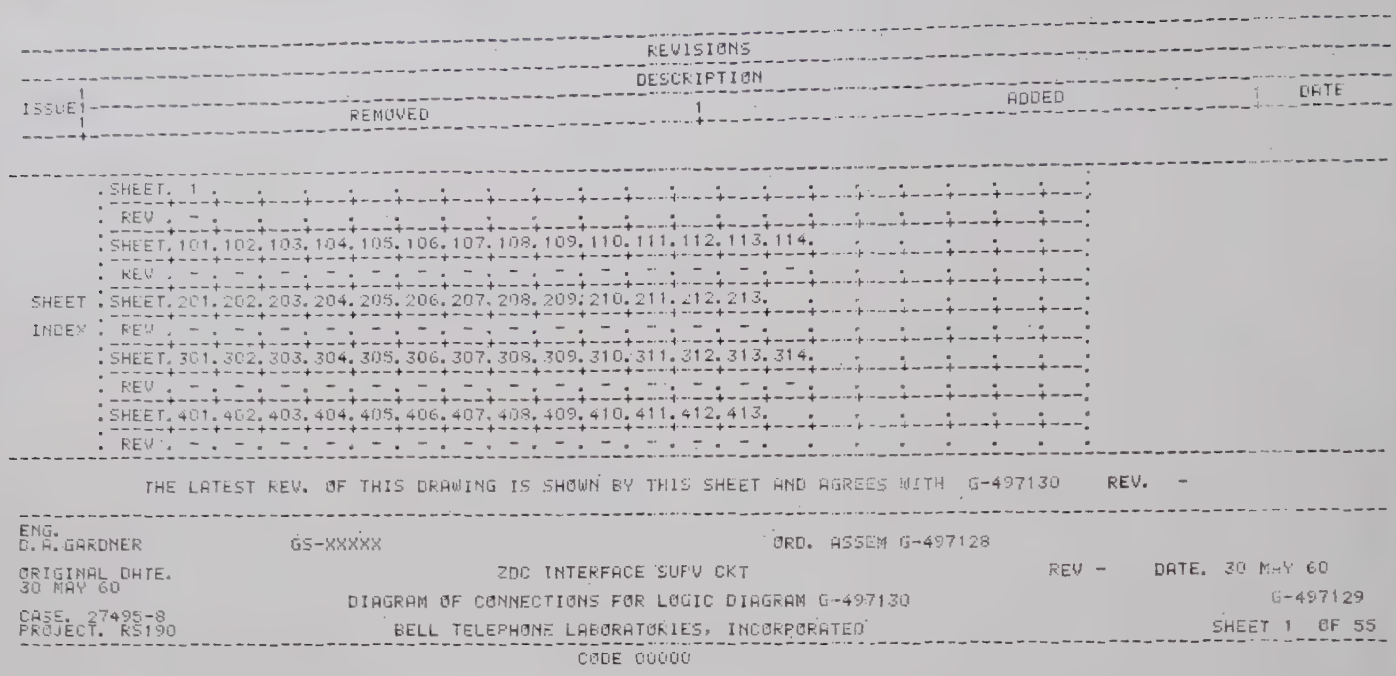


Fig. 8—Sample cover sheet prepared by DIPS. Space for revision data below the revision heading has been dropped from the figure.

descriptions to identify the unit involved. Note the three columns on the right, which indicate, respectively, the wiring configuration and the lengths to within 1/10 inch.

The Wiring Diagram Program

The order DGRAM will produce a set of figurative wiring diagrams for all the cards of a drawer.[4] The program considers one card at a time and prints out images of all wires connecting to this one card. Thus, 80 cards would require 80 diagrams. Wires which terminate in the same file as the particular card considered are shown completely. Wires which terminate in other files are brought to the edge of the drawing and their destinations labeled to identify the destination (TO) terminals. Titling information is added to the bottom of the sheet. Cover sheets containing a table of revisions, index and further titling are also produced. Figs. 7 and 8 show samples of wiring diagrams and cover sheets.

The Programming Structure

The DIPS routines are part of an interpretive system controlled by a monitor program which examines the order cards and withdraws the required routines from the library tape. Such a system is particularly valuable since it permits the user to select only the routines he needs at each stage of the processing of data. It also significantly reduces the interdependence of the routines. This is particularly important when progress in a development project necessitates some changes in one of the routines without requiring changes in all routines.

[4] D. B. Kirby and C. W. Rosenthal, "A Computer Program for Preparing Wiring Diagrams," AIEE Paper No. 60-1007; August, 1960.

Conclusion

The programming system we have described is the result of the combined efforts of a number of people over a period of nine months (see Acknowledgment below). As a result of this work, the DIP system can process information for one drawer from start to finish (including preparing tapes for the production of documents) in a period under 30 minutes. The DIPS programs form an interpretive system which contain over 15,000 instructions. By interpreting the specified instructions of the engineers, DIPS can rapidly, economically, and accurately introduce automation to the development process. DIPS is still only an introduction of machine aids techniques, since there are programs that need to be written for engineering tasks not assumed by DIPS (logic node assignment and interchassis wire running lists, among others). These tasks will be considered in the design automation programs now being prepared to aid the development of other digital systems.

Acknowledgment

While there are too many individuals to mention all, the author wishes to thank Mrs. L. S. Boginsky, Mrs. D. B. Kirby, J. S. Bomba, N. M. Haller, N. A. Martellotto, and N. Bronstein for their contributions.

Improvement of Electronic-Computer Reliability through the Use of Redundancy*

W. G. BROWN†, J. TIERNEY‡, MEMBER, IRE, AND R. WASSERMAN‖, MEMBER, IRE

Summary—**Physical elements used for switching logic have finite probability of failure. The application of redundancy to logic circuits is presented for improving computer reliability. This paper shows various redundant configurations considered and the conclusion drawn.**

From all of the considerations, the majority gate provides a practical method for increasing the reliability. It shows that for operating periods which are short compared to the mean time to failure of the elements, a substantial increase in system reliability is obtained with majority-gate redundancy.

I. Introduction

IN large complex digital systems where the mean free-time-to-failure is annoyingly small, the question of reliability arises. In large data processing systems, reliable operation may be obtained by allowing a certain amount of down time for marginal checking and systematic replacement of components. However, there are many applications where uninterrupted operation is a critical requirement for economic or strategic reasons. In those cases where the required time interval of unserviced operation is long, or the system complex, the reliability requirement may be impossible to meet. In the former case, component reliabilities have decreased with time sufficiently to decrease system reliability, and in the latter, system complexity demands a component of unrealistically high reliability to yield a reliable system.

Within certain reasonable bounds, the use of redundant elements in these digital systems can improve reliability substantially. In particular, a "majority" type of redundant design can improve the reliability of a logical net under certain reasonable physical restrictions.[1] For our purposes reliability can be defined as follows:

1) System reliability is the probability of successful uninterrupted functioning of a digital system over some finite time interval.
2) Component reliability is the probability of successful component operation over this same time interval. Both of these probabilities are functions of time, but appropriate values in the interval can be used.

* Received by the PGEC, October 18, 1960; revised manuscript received, February 2, 1961. This study was performed at Hermes Electronics Co., Cambridge, Mass., supported by the Office of Nav. Res. under Contract Nonr-2133(00). Original manuscript by J. Tierney and R. Wasserman, Hermes Rept. No. M-828, dated April 15, 1960.

† Cook Research Labs., Morton Grove, Ill. Formerly with Hermes Electronics Co., Cambridge, Mass.

‡ Raytheon Mfg. Co., Waltham, Mass. Formerly with Hermes Electronics Co., Cambridge, Mass.

‖ Hermes Electronics Co., Cambridge, Mass.

[1] J. Von Neumann, "Probabilistic logics," in "Automata Studies," Princeton University Press, Princeton, N. J., No. 34; 1956.

The definitions are naive, but useful.

The digital systems to which this paper is directed operate under the following conditions:

1) The operating interval is sufficiently long or the system is so complex that the probability of successful operation is less than that called for by the application.
2) The system is not serviceable during the pertinent time interval for economic, physical, or strategic reasons.
3) Equipment cost and size are of secondary design importance (although order of magnitude increases are certainly undesirable).

II. Basic Ideas

The use of redundancy to increase the reliability of digital systems is not new. Individuals have demonstrated that significant improvements are possible under certain reasonable restrictions on component reliability.

The question of what sort of redundancy should be used and at what level the redundancy is applied depends basically on what sort of element one is contemplating for use as the switching and checking device. If the same device is used for all basic logical building blocks (*i.e.*, for checking functions as well as checked functions), then all blocks have about the same reliability, and checking elements are far from ideal. However, considering that a large group of basic building blocks perform complex logical functions (*e.g.*, a shift register), and that this is checked with single block, the relative reliabilities indicate that the single checking element can be considered as ideal compared to the large numbers of blocks used in the register. This leads to a redundant system with few comparisons and self-corrections. On the other hand, using a very reliable checking element and a poor functional element, the check might as well be at the single unit level since the addition of a large number of high reliability checking functions will improve the system performance.

These are some of the intuitive statements one can make about redundant configurations. This paper is concerned with justifying the statements.

III. Analysis of Redundant Configurations

A. Nonredundant System R_0

The standard of comparison taken is the nonredundant system R_0 consisting of N universal logical elements f. Nonredundant means that successful operation of all the N elements is a necessary condition for the

successful operation of the system. The reliability of R_0 is

$$R_0 = P_1 \cdot P_2 \cdot P_3 \cdot \cdots Pi \cdots P_N = \pi_1^N Pi, \quad (1)$$

or, if all the elements have about the same reliability P, then

$$R_0 = P^N. \quad (2)$$

Considering a configuration of 100 elements, each with a reliability of 0.99, the system reliability R_0 is only 0.37. This is precisely the reason for consideration of redundant systems. For as $N \rightarrow \infty$, $R \rightarrow 0$ regardless of how good P is, $0 \leq P < 1$.

B. Simple Redundant Configuration

Assuming that a perfect decision and switching element exists,[2] one approach to increasing system reliability is parallel duplication of functional units and an associated switch to sample the redundant outputs and to make a choice between the outputs [Fig. 1(a)]. For a functional element with a probability of success P, the redundant configuration has a reliability of $1-(1-P)^2$, assuming the decision element is perfect and the redundant block functions if either function element operates. For a redundant configuration of M function elements [Fig. 1(b)], the reliability is $R=1-(1-P)^M$, and for a system of N redundant blocks the reliability $R=[1-(1-P)^M]^N$. The reliability for this configuration can be made as close to unity as desired, regardless of the complexity N by increasing M. In fact $R \rightarrow 1$ as N, $M \rightarrow \infty$.

Another redundant approach using an ideal decision element is shown in Fig. 1(c), where the ideal element checks several system groups for a correct output. The reliability for this configuration is $R=1-(1-P^N)^M$, and M does not have the same control over the reliability as it did in the configuration shown in Fig. 1(b), for $R \rightarrow 0$ as N, $M \rightarrow \infty$. Actually it is obvious from the two configurations that Fig. 1(b) is a more efficient configuration than Fig. 1(c), since Fig. 1(b) has many more modes of allowable element failures without system failure. At least for the ideal checking and switching element then, low-level checking is more efficient, as mentioned in Section II.

For the switch to distinguish the correct output, there must be some additional information available. A method for motivating the decision switch in choosing a correct output from a redundant group is to build the redundant groups out of an odd number of function units and use a simple majority to decide upon the correct output.

[2] F. Moskowitz and J. B. McLean, "Some reliability aspects of systems design," IRE Trans. on Reliability and Quality Control, vol. RQC-8, pp. 7–35; September, 1956.

The first class of redundant configurations (Fig. 2), is constructed from system R_0 by partitioning R_0 into the logical function units f and replacing each function unit with three function units and a single majority gate with a threshold set at 2. Therefore, the only way this redundant block fails is if more than one function f unit fails, assuming the majority gate is perfect. The reliability of this configuration is

$$R = P^3 + 3(1-P)P^2 = P^2(3-2P). \quad (3)$$

For a system consisting of N of these building blocks,

$$R_1 = [P^2(3-2P)]^N. \quad (4)$$

For example, if $N=100$ and $P=0.99$, then $R_1=0.97$. Notice, however, that for each nonredundant unit in a simple system, three function boxes and a majority gate are now needed leading to a fourfold increase of equipment.

The relative improvement in reliability offered by this configuration over R_0 can be determined by plotting the ratio

$$\frac{R_1}{R_0} = \frac{[P^2(3-2P)]^N}{P^N} = [P(3-2P)]^N \quad \text{for } 0 \leq P \leq 1 \quad (5)$$

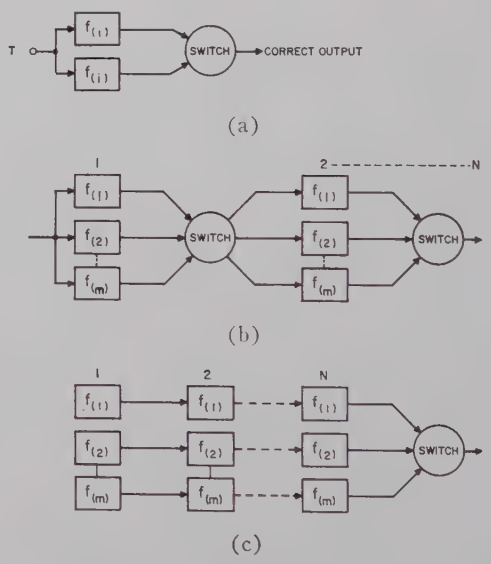


Fig. 1—Simple redundant configurations.

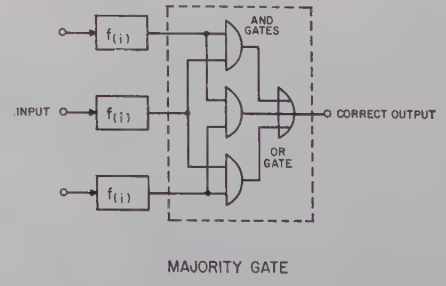


Fig. 2—Single majority-gate output (system R_1).

selected for N as in Fig. 3. It is of interest to note that even in this highly idealized case, the reliability of a system R_0 cannot be improved by type R_1 redundancy unless $P>0.5$. For the remainder of the investigation, then, values for P can be restricted to the range $0.5 \leq P \leq 1.0$.

To make the building block of Fig. 2 more realistic, consider a majority gate with some finite probability of failure, *i.e.*, a reliability $m<1$. By introducing m into (4), the reliability of this system is then

$$R_2 = [mP^2(3 - 2P)]^N. \quad (6)$$

For values of $m=P$,

$$R_2 = [P^3(3 - 2P)]^N = P^N[(P^2(3 - 2P)]^N,$$
$$R_2 = R_0\,[P^2(3 - 2P)]^N = R_0R_1 \quad (7)$$

Since R_1 is always <1 for physical elements, R_2 can never be as good as R_0 (the reliability of the nonredundant system). However, instead of considering a majority operation after every triplicated function box, consider a group of K functions cascaded before a majority is sampled, as in Fig. 4. With such an arrangement it is possible that the number of majority gates could be found which maximizes $R_2>R_0$. By introducing the variable K (the number of function boxes cascaded with each majority gate), into (6) and (7), R_2 becomes

$$R_2 = [mP^{2K}(3 - 2P^K)]^{N/K}. \quad (8)$$

To simplify analysis, assume $m=P^L$ where $0 \leq L < \infty$. Then

$$R_2 = [P^{(2K+L)}(3 - 2P^K)]^{N/K}. \quad (9)$$

Maximizing R_2 with respect to K yields an implicit relation for K,

$$P^L = (3 - 2f)^{-1}\frac{-(2f)}{f(3 - 2f)}, \quad \text{where} \quad f = P^K. \quad (10)$$

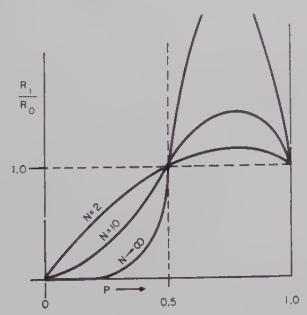


Fig. 3—Relative reliability of the single majority-gate output system.

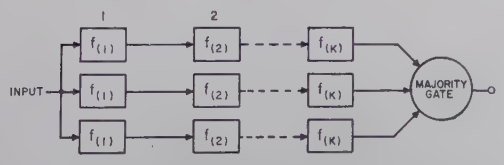


Fig. 4—System R_2 with K functions cascaded before a majority checking operation

This equation can be solved systematically for K and the corresponding R_2 obtained. Note that the values of K obtained are not integral in general, and an approximating integer must be chosen to implement the physical system. For the case of $L=0$, the perfect majority gate, the system reliability R_2 is better than the reliability of R_0 over the entire range $0.5<P<1$. $R_2=R_1$ for $K=1$, and R_2 is a monotonically decreasing function of K as in Fig. 5. The value of K to maximize R_2 and at the same time be physically realizable is $K=1$. This result is what is intuitively felt to be correct; namely, for perfect majority gates, sampling at the lowest level of equipment complexity gives the best result.

For the general case when the reliability of the majority gate is P^L (some multiple of function probabilities), several curves have been plotted in Fig. 6. (The Nth root of all reliabilities has been shown.) For R_2, K equals K max:

$$[R_0]^{1/N} = P \quad (11)$$
$$[R_1]^{1/N} = [P^2(3 - 2P)] \quad (12)$$
$$[R_2]^{1/N} = [P^{(2K+L)}(3 - 2P^K)]^{1/K}. \quad (13)$$

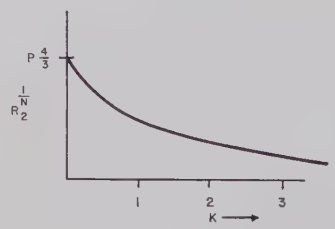


Fig. 5—Relative reliability of system R_2 using a perfectly reliable majority gate.

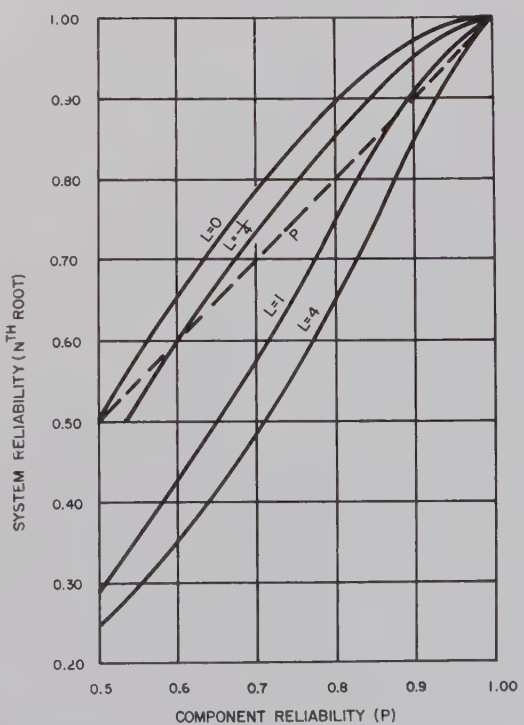


Fig. 6—System R_2 reliability for component reliability (P).

Notice that K max is a function of P. Also, as $L \to \infty$ and $K \to \infty$, $R_2 > R_0$ becomes smaller even for $K = K$ max. In fact for any $P < 1$ there is an $L < \infty$ such that no K exists to make $R_2 > R_0$.[3]

C. Two-out-of-Three with Majority Gates in Triplicate

The single majority is interesting, but, for improving a single imperfect majority gate on the output of a trio, the most obvious next step is to improve majority gate reliability by triplicating majority gates on the outputs of the function boxes, as shown in Fig. 7. Unfortunately now the basic configuration is quite practical but analytically complex. In the case of one majority gate for every building block, the gates performed an isolation function, so that the only way an error could propagate was for a building block to fail independently of any other block, regardless of how complex the functional units become. A typically interconnected unit is the comparator. The logical diagrams for this unit are shown in Fig. 8 for both the nonredundant and the redundant case with the ideal majority gate introduced on the lowest level. Notice in Fig. 7(b) that the only way a computer failure can occur is for one of the redundant boxes (dotted lines) to fail. There is no interaction at all between boxes. On the other hand, consider the comparator circuit of Fig. 9 (built from redundant building blocks of the type shown in Fig. 7), where the majority gates can fail and there are three majority gates for each building block. Notice that in addition to the configuration failing for any one of the three redundant blocks failing, several other critical conditions can cause failure. For example, majority gate 1 in the upper left unit can fail causing the drive to the number 1 OR gate to fail and consequently the output from this OR gate may be in error. At the same time, number 2 OR gate could fail, invalidating its output; thus two-out-of-three outputs from this group would be erroneous and the computer would fail. It is important to note that the individual building blocks can still function properly in Fig. 9 (*i.e.*, single errors), but cause a system failure. In other words, the reliability of a single redundant building block taken by itself loses meaning in this case, and we have to approach the analysis of this model from another direction. In order to evaluate the triplicate majority gate on output packages, a specific interconnection of blocks must be considered. However, for the moment consider first the majority gates on input lines.

[3] R. Miller, "Majority Logic Analysis," Hermes Electronics Co., Cambridge, Mass., Rept. No. M-895; August 15, 1960.

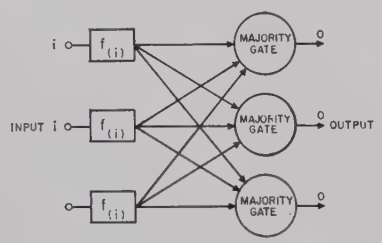
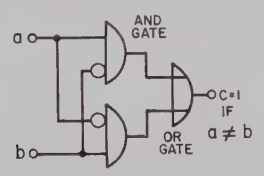
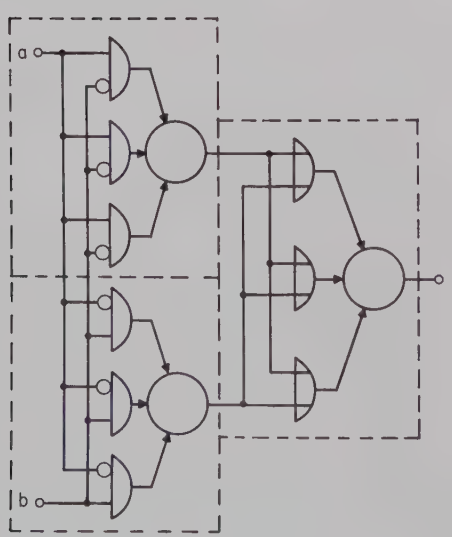


Fig. 7—Majority gates in triplicate.



(a)



(b)

Fig. 8—Bit comparator.

D. Majority Gates on Input Lines

This basic package, shown in Fig. 10, results from

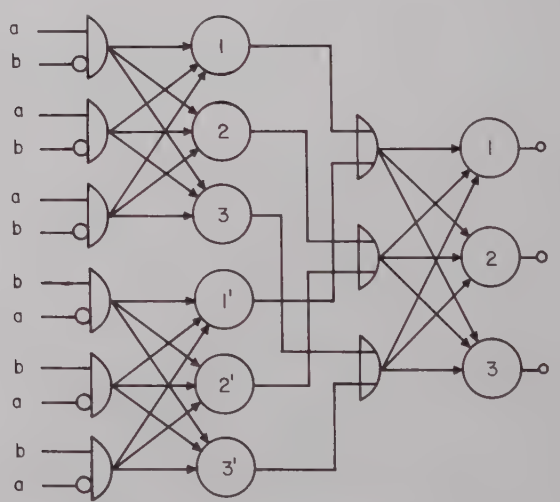


Fig. 9—Redundant bit comparator.

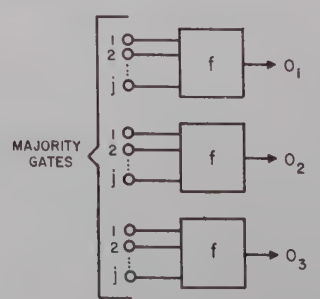


Fig. 10—Majority gates on input lines.

associating the majority checking and restoring gate with each input rather than output. Such a variation in the way we construct the redundant block is a trivial one, except for the cases of branching on function outputs. In Fig. 9, we can visualize forming this new package by associating all of the left-hand majority gates with the right-hand function trio, and so on throughout the system. Note that this leads to the same number of majority gates only in the case of no output branching. As soon as branching occurs, more majority gates must be added to form this input checking block. The addition of majority gates in this case is the price paid for the independent block. Considering a functional unit with j input lines, as shown in Fig. 10, and majority gate checking on these input lines, each package of a functional unit and input gates can be considered independently. With function success probability P, and gate probability P^L, each package's probability of success equals $r=P^{Lj}P$.

The reliability of the configuration of Fig. 10 is, then, the probability that all three packages function, or that two function. Thus

$$R = r^2(3 - 2r), \tag{14}$$

$$R = (P^{Lj}P)^2(3 - 2P^{Lj}P), \tag{15}$$

and, for a system of N blocks,

$$R_3 = [(P^{Lj}P)^2(3 - 2P^{Lj}P)]^N. \tag{16}$$

We now have the extra factor j in this expression since we are considering these functional units from the point of view of input lines. The question again arises, "How many levels of functions shall we check with a level of majority gates?" Once again it is a question of balancing the reliability of the majority element against that of the functions it is checking. Since the number of input lines to a composite group of functions determines the total of majority gates used, we must consider the manner in which a group of K boxes would be grouped together and the resulting number of input lines. A reasonable approach is to introduce a new parameter s, the number of inputs "saved" (*i.e.*, not brought out externally), when several functions are clumped together, for a majority check. For example, in Fig. 11(a), $j=3$, $s=2$ and, in Fig. 11(b), $j=2$ and $s=1$, parameter s may range from 0 to j, and we are assuming j is constant for all the blocks in a group of K. The total number of inputs with majority gates on them becomes

$$i = K(j - s) + s,$$

for j inputs to each function box, s saved on combining two, and a total of K combined. The only restriction we have is the assumption of a uniform j throughout the system logic. This is not unreasonable, and, in fact, for a system built of a basic two-input function, this is the case. If we use this general expression for input lines to a group of K boxes, we have for R_4,

$$R_4 = [P^{2(K[Lj-Ls+1]+Ls)}(3 - 2P^{K(Lj-Ls+1)+Ls})]N/K, \tag{17}$$

or

$$R_4 = [Q^2(3 - 2Q)]^{N/K},$$

where

$$Q = P^{K[Lj-Ls+1]+Ls}.$$

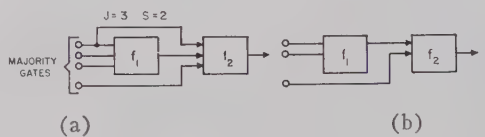
Once more maximizing with respect to K, we obtain the implicit equation as follows:

$$P^{Ls} = Q^{-\frac{Q}{3-3Q}}(3 - 2Q)^{-\frac{3-2Q}{6-6Q}}, \tag{18}$$

and the supplementary relationship

$$K \max = \frac{s}{j - s + \frac{1}{L}}\left[\frac{\log Q}{\log P^{Ls}} - 1\right]. \tag{19}$$

Assuming a specific j, s, L and Q, one solves for P in the first relation and determines K max from the second. In Fig. 12 $[R_3]^{1/N}$ is plotted for $j=2$, $s=1$, $L=1$, $K=K$ max, and $K=1$ just to indicate the effect of a realizable net with constant K. Notice that these curves are below the nonredundant reliability for a large range of component P. However, in the range



(a) (b)

Fig. 11—Examples illustrating input "saving."

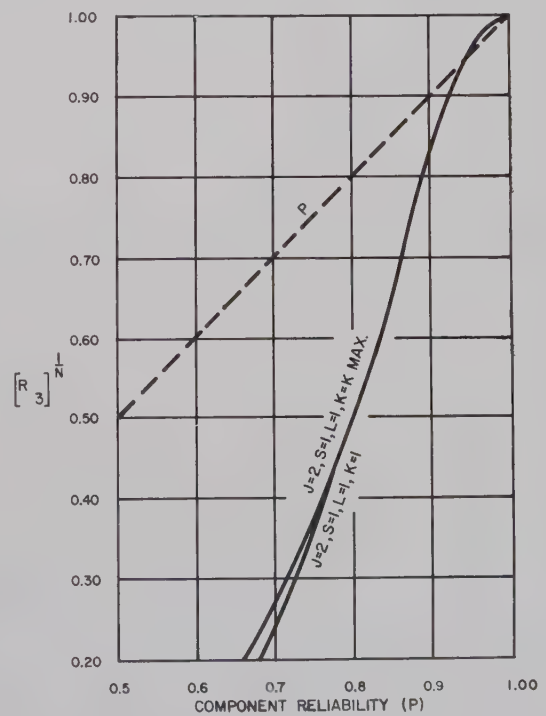


Fig. 12—Relative reliability of system R_3.

above $P=0.9$, this function exhibits very high reliability as we shall see.

Since all of these models exhibit their higher reliabilities in the region above a component probability of about 0.9, let us restrict ourselves exclusively to this region for component reliability. This is not unrealistic for components used in present-day digital systems. As a function of time, however, these component reliabilities tend to decrease. Consequently, for those systems which must operate for periods so long that component reliability decreases to below about 0.9, these configurations add little reliability and, in fact, can reduce reliability.

E. Reliabilities for P Near Unity

We may expand our reliability expressions in a power series about the point $P=1$ so: $R(P)=A_1(1-P)+A_2(1-P)^2$, and, since $(1-P)$ is the probability of a single element failing and, in general, $(1-P)^K$ is the probability of K elements failing, we can interpret the coefficients accordingly. For a triplet majority gate scheme, no single element failure causes system failure, and so $A_1=0$. A_2 is then the number of ways two elements failing can cause a system failure. Since we are interested in $P>0.9$, we can truncate the series at this term and we have $R(P)=1-K_2(1-P)^2$, where K_2 is the number of "critical pairs" in the system (*i.e.*, the number of failed pairs causing system failure). For the single majority gate on output, we can use $R(P)=1-K_1(1-P)-K_2(1-P)^2$ since a single failure of the majority element can also cause system failure.

In general, it can be shown that these approximations are lower bounds on the reliability for all P, and are equal to the reliability for P near 1. Using these series approximations, and considering the ratio $1-P/1-R^{1/N}$, or component unreliability to system unreliability (normalized), the following curves result. For the triplicated functional units with one ideal gate on output we have

$$\frac{1-P}{1-R_1^{1/N}}=\frac{1}{3(1-P)}. \tag{20}$$

For the triplicated functional units with one nonideal majority gate checking output,

$$\frac{1-P}{1-R_3^{1/N}}=\frac{K}{L+3K^2(1-P)} \tag{21}$$

with K and L as previously defined and for the case of $K=K$ max

$$\frac{1-P}{1-R_3^{1/N}}=[12L(1-P)]^{-1/2} \tag{22}$$

where

$$K\ \max=\left[\frac{L}{3(1-P)}\right]^{1/2}.$$

For the more useful configuration of majority gates on the inputs to three functional blocks the ratio is

$$\frac{1-P}{1-R_4^{1/N}}=\frac{K}{3T^2}(1-P)^{-1}$$

with

$$T=L\left[K\left(\frac{j-s+1}{L}\right)+s\right] \tag{23}$$

and

$$K\ \max=\frac{s}{j-2+\dfrac{1}{L}}.$$

All of these curves have been plotted on log-log paper and are presented in Fig. 13. For all of these curves, L is set equal to unity (the majority gate is taken to have the same reliability as the elements it is checking).

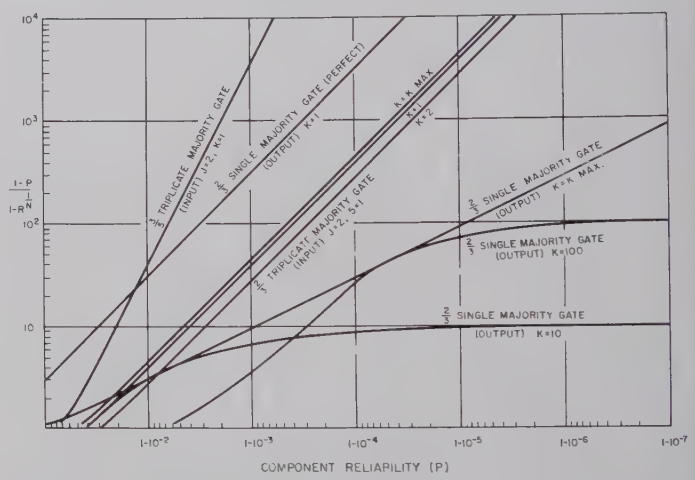


Fig. 13—System reliability for component reliability near unity.

Curves for higher-order majorities with majority gates on input lines can be shown to have the general relation

$$\frac{1-P}{1-R^{1/N}}=\frac{K}{C_N^{2N-1}T}[T(1-P)]^{-(N-1)}; \tag{24}$$

T as before where N out of $2N-1$ is the majority level, and C_N^{2N-1} is the combination of $2N-1$ things N at a time. The curve for a 3 out of 5 majority configuration has also been plotted.

It is worth noting that in this high probability region the relation

$$K\ \max=\frac{s}{j-s+\dfrac{1}{L}}$$

yields values less than unity for practical values of j, s, and L, and we would choose $K=1$ to implement this redundancy.

IV. Triplet Majority Checking on Output Lines

Having examined the various majority logic schemes presented and with Fig. 13 as a design guide, let us re-examine the configuration introduced earlier, the triplet majority gate on output (Fig. 7).

As was pointed out in connection with Fig. 9, the triplet majority gate on output can be converted to a majority gate on input simply by associating the majority gate checking with input lines rather than output. However, as soon as branching of an output line occurs, the majority gate on input scheme requires more components (majority gates).

The output checking configuration reliability (*i.e.*, Fig. 7) depends to some extent on the logical net in which it is placed, and the extra majority gates needed. Our previously analyzed majority gates on the input configuration suggests that the output checking package may actually be more efficient.

The evaluation of any majority output checking block requires a specific logical net because of the interdependence of building blocks. However, we can consider a very tightly connected net, such as Fig. 14 and evaluate the reliability for $P>0.9$. Fig. 14 indicates a symmetrical chain connection of majority-gates-on-output building blocks. Each building block consists of a trio of majority gates on the outputs of a trio of function boxes (in the same fashion as Fig. 7), each having j inputs and consequently being fed by other j blocks in the chain (with one or more feedback loops if not trivial). Each block or circle then represents a trio of one type. The simplest nontrivial example of one of these chains is given in Fig. 15(a). For purposes of analysis, it is convenient to redraw the chain as in Fig. 15(b), assuming that the connection from function output to majority gate exists, although no longer shown. These chains may grow longer with N and stay fixed in terms of input complexity J, or they may increase in both N and J. As in the previous sections, the reliability of such a chain may be determined by tabulation of the critical pairs which can occur. Starting with simpler chain and working up to more complex ones, and considering the majority gate reliability to be $m=P^L$ and the component reliability to be $P>0.90$, an expression is obtained for chain reliability R in terms of N, j, and L:

$$R_5 = 1 - 3N[(L_j + 1)^2 - L^2(j-1)^2](1-P)^2; \quad N \geq 2j - 1. \tag{25}$$

Taking the Nth root of these expressions, we have an equivalent average reliability for a building block of this type to compare with previous expressions derived for individual independent building blocks. Thus,

$$R_5^{1/N} = 1 - 3[(Lj + 1)^2 - L^2(j-1)^2](1-P)^2; \quad N \geq 2j - 1. \tag{26}$$

Fig. 14.

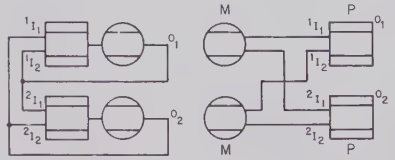


Fig. 15.

In order to compare this model with a nonredundant building block of success probability P, consider

$$\frac{1-P}{1-R_5^{1/N}} = \frac{1}{3[(Lj+1)^2 - L^2(j-1)^2]}(1-P)^{-1}; \quad N \geq 2j - 1. \tag{27}$$

From the previous section, the comparable ratio for the independent majority gate case is

$$\frac{1-P}{1-R_4^{1/N}} = \frac{1}{3(Lj+1)^2}(1-P)^{-1}. \tag{28}$$

It is obvious by examining the expressions that (28) is always less than (27), and that they are equal only for the trivial cases of $L=0$ or $j=1$. At any rate, the average building block reliability we have obtained from an analysis of this interconnected chain of redundant elements with majority checking on output is always better than a completely independent input majority checking building block.

The notion of long groups of cascade functional units checked by a single majority level is not discussed, for it turns out that the special case of K equals unity is sufficient to establish the inequality between the two systems. For K greater than unity, the input gate model only adds more majority gates to its configuration, whereas the output gate case does not. Any change in probabilities can be absorbed by the parameter L.

For the long chain of higher than 2 out of 3 majority, a general expression for the critical group of failures is

$$A_m = C_m^{2m-1}[(Lj+1)^m - [L(j-1)]^m], \tag{29}$$

with $R=1-A_m(1-p)^m$, and

$$\frac{1-p}{1-R^{1/N}} = \frac{1\ (1-p)^{-(m-1)}}{C_m^{2m-1}[(Lj+1)^m - [L(j-1)]^m]}. \tag{30}$$

Fig. 16 indicates the $1-P/1-R^{1/n}$ curves for $j=2$, $L=1$, and 2 out of 3, 3 out of 5, 4 out of 7 redundancies. If we compare these with Fig. 13, we see that the two configurations (multiple majority checking on input or on output), are comparable.

It turns out that, while the symmetric chain configuration is very closely interconnected, the number of

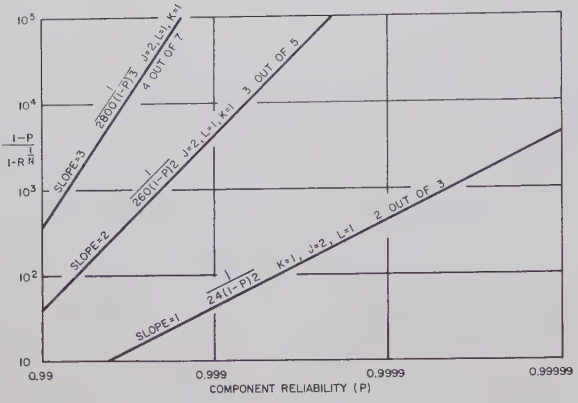


Fig. 16—Reliability of redundant symmetric chain.

critical failure pairs that can occur is less than the maximum possible for a configuration of packages with majority gate checking on output lines. In fact, because of the symmetric chain configuration, the reliability turns out to be close to the maximum possible with output majority checking. However, it is shown in the Appendix, that even a worst case of output majority checking is still better than a comparable input checking package.

V. Conclusion

This paper has been decidedly a microscopic approach to redundant logical nets. This is the inevitable result of trying to talk about physical configurations and practical checking schemes. Fortunately, if we are willing to consider components with high reliabilities (*i.e.*, $P>0.9$),[4] we can use a Taylor series about $P=1$ to represent all our configurations, and, what is more, can reduce the series to two useful terms, one of which is a constant, and the other a factor whose coefficient we can physically interpret as the critical n' tuplets of a system (*i.e.*, the number of groups of n components whose failure causes system failure). Using this approach, a reasonable microscopic examination becomes possible.

The curves in Figs. 13 and 16 result from such an approach, and from them we can consider implementing various redundant configurations as discussed. As for the choice of configuration, it is obvious from our previous discussions that the package with majority gates on output is the most feasible.

In many cases, it may be wise to consider a higher-order majority and fewer checking functions (*i.e.*, $K>1$). In all cases, the higher-order majority introduces large gains in reliability, decreased slightly by an increase in K. At the same time, the increase in K reduces the amount of equipment in the higher-order system to a comparable level. For example, the equipment in a 3 out of 5 majority checking system with $K=5$ is about the same as a 2 out of 3 system with $K=1$.

[4] Von Neumann and others have shown that no strong statements about redundant nets can be made unless component $P>0.9$ approximately.[1]

The conclusions drawn from this microscopic approach to redundant logical design led to the development of a majority gate module as shown in Fig. 7. From all of the considerations, this majority gate provides a practical method for increasing the reliability of digital systems. The first trio f is composed of the functional units of the module. The second trio m are the majority gates which perform the two-out-of three comparison.

The nature of the majority gate suggested that a technique involving the principle of summation of some physical quantity would be the ideal element. Logical decision would then depend on whether the sum of the input quantities exceeded a preselected threshold level. Varying the threshold level permits changes in the logical function, and this property is exhibited in magnetic cores,[5,6] parametrons,[7] and other similar devices.

Appendix I

A reasonable way to compare our various redundant configurations is to talk about the possible critical pairs that each configuration introduces according to the discussion of Section III.

We can identify four kinds of critical pair occurrence defined as follows:

f^2 = two function units failing,
m^2 = two majority gates failing in a single package,
mf = a single function unit and a single majority gate failing,
mm = two majority gates failing in separate packages.

For the majority gate on input package, the type-mm failure does not occur since they are all absorbed in the m^2-type failure. However, we can break up the m^2 failures into an mm group and an m^2 for the purposes of comparison.

In general, for an output checking package, there could be a maximum of $3j(j-1)$ mm-type failures (j in the number of inputs), although we would expect much less since many pairs occur in several places and would be counted only once. Table I compares the three configurations.

TABLE I

Failure Type	1) Input Majority	2) Output Majority	3) Symmetric Chain
f^2	3	3	3
m^2	$3j$	3	3
mf	$6j$	$6j$	$6j$
mm	$3j(j-1)$	$3j(j-1)$ max	$6(j-1)$ average
Total critical pair coefficient	$3(j+1)^2$	$3(j+1)^2-(j-1)]$	$3[(j+k)^2-(j-1)^2]$

[5] R. Wasserman, "Development of a Majority Gate for Improving Digital System Reliability," Hermes Electronics Co., Cambridge, Mass., Rept. No. M-809; November 2, 1959.
[6] R. Wassermann, "Summing core logic yields digital circuit flexibility," *Electronic Design*; October 12, 1960.
[7] D. Sims, "The Parametron Used as a Majority Gate for Improving Digital System Reliability," Hermes Electronics Co., Cambridge, Mass.; September 15, 1960.

From this table, it is clear that the symmetric chain configuration yields a lower number of critical pairs (*i.e.*, a higher reliability) than the general worst-case output majority package, but both of these are better than the input package.

Appendix II

Derivation of the Symmetric Chain Reliability Expression

The expressions for the reliability of the symmetric chain structures may be obtained in a direct fashion. We may expand the reliability expression in a power series about the point p equals one. Thus,

$$R(p, L, N) = A_0 + A_1(1 - p) + A_2(1 - p)^2 + \cdots A_n(1 - p)^n + \cdots, \qquad (31)$$

where the A's may be functions of L and N, N=length of chain, L is defined as $m = p^L$, m=majority gate probability. If the coefficients are examined, $A_0 = R(1, L, N) = 1$. Since $(1-p)$ is the probability of a single element failing, and $(1-p)^k$ in general is the probability of k elements failing, the coefficients of the series represent the number of ways, 1, 2, $\cdots$, k elements can fail and cause a system failure. $A_1 = 0$ since no single element can cause system failure. A_2 then is the number of ways two elements failing can cause a system failure. The higher coefficients can be interpreted in similar fashion, but A_2 is the last coefficient of concern at this time. Interested in element probabilities of success, that are very close to unity ($p = 0.90$), so that the power series may be truncated at A_2.

$$R = 1 + A_2(1 - p)^2 = 1 - K(1 - p)^2, \qquad (32)$$

where K is the number of "critical pairs" in the configuration of concern. Before the symmetric chain is examined, consider the majority gate reliability m. In general, use $m = p^L$, for p close to one, $(1-m) = [1-(1-(1-p))^L] = 1-1-L(1-p)$.

$$1 - m = L(1 - p). \qquad (33)$$

There are two distinct conditions for a chain of the type considered. For $N \geq 2j-1$, critical pairs occur differently than for $j \leq N \leq 2j-1$ ($N < j$ is a degenerate case). For calculation consider $j = 2$ and the two chains of Fig. 17, with $N = 3$, and $N = 5$ ((a) and (b) represent the different cases). The critical pairs can be divided into several types.

1) Pairs occurring in a single function trio.
2) Pairs occurring in a single majority trio.
3) Pairs occurring from interaction of single majority failures in different trios.
4) Pairs occurring from one majority gate failure and one function box failure.

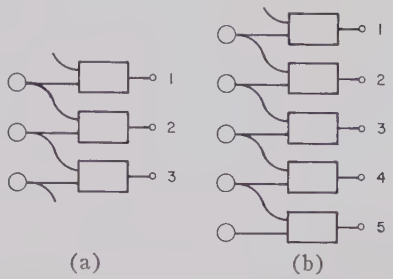


Fig. 17—Symmetric chain.

Notice that there is no critical pair production in the different function trios comparable to (3) for majority gates. This is due to the isolating effect the majority gates have on single function failures. For both the long and short chain, 1), 2), and 4) are the same and are determined as follows looking at Fig. 17.

1) For a single function trio a critical pair can occur in C_2^3 or 3 ways. Since there are N groups, then $3N(1-p)^2$ is the corresponding probability of occurrence.
2) For a single majority trio, the pairs occur 3 ways or a total of $3N$. So probability $= 3NL^2(1-p)^2$.
3) For a single majority error and a single function error, a single majority error can occur in 3 ways for any trio and a function box error can combine with it in $2j$ ways. For a chain N in length, the probability is $6Nj(1-p)(1-m) = 6LNj(1-p)^2$.

For failures caused by condition 3) (interaction of majority failures in different groups), each chain must be considered separately. The shorter chain ($j \leq N \leq 2j-1$) has a completely dependent critical pair grouping for the type-3 error. If one majority gate fails, there are always two gates in each of the other trios which can fail and cause a critical pair as long as $N \leq 2j-1$. This can be seen in Fig. 17. In this situation the critical pairs equal $3 \cdot 2C_2^N$ or $3 \cdot N_2(N-1)$. The probability associated with this is $3 \cdot N(N-1)(1-m)^2 = 3L^2N(N-1)(1-p)^2$. For the long chain, $N > 2j-1$ there are some majority trios in which a failure in any of the three gates can still be tolerated after a gate has failed in some other trio. An example of this is the majority trio 5 shown in Fig. 17, after one of the majority gates in 2 has failed. For the long chain the number of critical pairs of this type are $3 \cdot 2 \cdot N \cdot (j-1)_2 = 6N(j-1)$, and the associated probability is $6N(j-1)(1-m)^2 = 6L^2N(j-1)(1-p)^2$.

Now add the separate critical pairs to get the second term of (32).

For the short chain $j \leq N \leq 2j-1$,

$$K = 3N + 3NL^2 + 6LNj + 3L^2N(N-1)$$
$$= 3N + 6NLj + 3L^2N^2$$
$$= 3N + 6NLj + 3L^2N^2 = 3N[1 + 2Lj + L^2N]$$
$$R = 1 - 3N[1 + 2Lj + L^2N](1 - P)^2. \qquad (33a)$$

For the long chain $N \geq 2j-1$,

$$K = 3N + 3NL^2 + 6LNj + 6L^2N(j = 1)$$

$$= 3N[1 - L^2 + 2Lj + 2L^2j]$$

$$= 3N[(L + j)(1 + 2Lj - L)]$$

$$R = 1 - 3N[(L + 1)(1 + 2Lj - L)](1 - P)^2 \quad (33b)$$

These arguments are easily generalized for high-order majorities.

ACKNOWLEDGMENT

The authors wish to thank Dr. Marshall Yovits of the Office of Naval Research for many stimulating suggestions, to acknowledge the early contributions of J. E. De Turk, and to thank Dr. E. Arthurs and J. McReynolds for their suggestions.

Some Thoughts on Digital Components and Circuit Techniques*

ARTHUR W. LO†, SENIOR MEMBER, IRE

Summary—**Signal standardization and control directivity are emphasized as the basic physical requirements in considering components and circuit techniques for the handling of digital information. The significance and the ways and means of meeting these requirements are most revealing when illustrated by the operations of the parametric phase-locked oscillator and the tunnel diode. A categorical listing of digital-gain elements, accompanied by illustrative comments, is presented to offer a unified viewpoint on digital components and circuit techniques in connection with present-day practice and prospective future development.**

I. INTRODUCTION

PRESENT-DAY digital systems, notably the electronic data-processing machines, employ components and circuit techniques largely carried over from conventional communication systems. While this approach has been practical and effective, the ever-creasing demand for digital machines of higher speed, larger capacity, more flexibility, and improved reliability stresses the need, among other things, for a fresh and broad view of physical components and circuit techniques particularly suitable for handling digital information. This paper is an attempt to formulate some general concepts concerning the physical requirements in networks that handle digital information, and the ways and means of realizing such networks with presently available and prospective components. While a broad viewpoint, not biased by conventional approaches, is taken here, physical realization is emphasized over pure reasoning. Illustrative examples are provided whenever a concept in general terms becomes vague and ineffective. Some generalization and deduction, not completely worked out, are also included to provoke thought and to invite discussion.

II. BASIC PHYSICAL REQUIREMENT

Since the purpose of digital networks is to handle digital information, one may well begin by examining the physical nature of digital information. The digital information contained in the simplest network (let us call it a digital cell) is physically recognizable by the *state* of the network (usually that of a network element) and by the *signal* derived from the network. This dual representation concept is a useful one, because information handling can be more easily visualized, in some cases, as manipulation of signal combination, and, in other cases, as manipulation of network combination.

In binary digital systems, a digital cell may assume one of two distinct states; the corresponding signal appears in one of two discrete and, nevertheless quantitative, measures of some physical quantity. In common with communication systems where information is physically represented by amplitude, frequency, or phase modulation of an electric quantity, digital signals can appear in *amplitude*, or *frequency*, or *phase* script, observed at certain locations in a certain time interval. The amplitude script is extensively used in present digital systems; but the others, particularly the phase script, should not be excluded in our general discussion.

The logical operation of a digital network can be specified in well-defined mathematical terms. However, the physical operations a network must perform to achieve such logical operations are difficult to specify. Nevertheless, one can express the basic and pertinent physical operations required in a digital network in

* Received by the PGEC, June 2, 1960; revised manuscript received, May 8, 1961. Part of this work was sponsored by the U. S. Navy Bureau of Ships.

† Components Division, IBM Corp., Poughkeepsie, N. Y. Formerly with RCA Labs., Princeton, N. J.

terms of *signal standardization*, *control directivity*, and *signal manipulation*.

A. Signal Standardization

Any sizable digital system contains a large number of nominally identical digital cells (or several sets of them) interconnected in a complicated manner. To insure proper operation without looking into every cell and tailoring every circuit to fit individual needs, all information signals are standardized to certain sets of quantitative specifications before signal deterioration (which is unavoidable in any practical system) grows to such an extent as to cause ambiguity.

Any incoming binary signal (before it is allowed to deteriorate beyond recognition) is meant to be either a "0" or a "1" signal, and the first requirement of a digital network is to recognize the signal through some form of *threshold detection*. Once the "0" or "1" nature of the incoming signal is established, the network acts accordingly and discriminatively to restore the signal towards its "0" and "1" standard form. For digital signals in amplitude script this process implies discriminative amplification and attenuation to "compress" the signal amplitude toward one of two prescribed values. For phase or frequency script signals the operating network acts to "pull" the signal discriminatively toward one of two reference phases or frequencies. It is to be noted that, for digital signal of any script, signal standardization requires *gain* in a network to compensate for dissipation in circuit elements and to enable one digital cell to drive more than one identical cell (which is often referred to as fan-out). Signal amplification *per se* is meaningless in digital data-processing systems; however, it is a most important factor in achieving signal standardization. For amplitude script signals, the amplification is discriminative with respect to the information-carrying quantity (amplitude). For phase or frequency script signals, signal amplification (to compensate dissipation and fan-out) is nondiscriminative with respect to the information-carrying phase or frequency.

The process of signal standardization is illustrated in Fig. 1, showing the relation between input and output signals of a digital network while interconnected with other networks. Fig. 1(a) shows the characteristic of an ideal network with perfect threshold and discriminative convergence characteristics; here, a great spread of input signals (caused by deterioration) can be tolerated. In practical cases, the threshold is not perfectly sharp, and, more significantly, the threshold does not occur at exactly the same location among supposedly identical elements. The practical situation illustrated in Fig. 1(b) indicates the need of greater separation of input "0" and "1" signals (or, rather, the need of signal standardization before much signal deterioration). One may say component uniformity is a most significant factor in considering digital networks.

Fig. 1 is a "normalized" representation, indicating

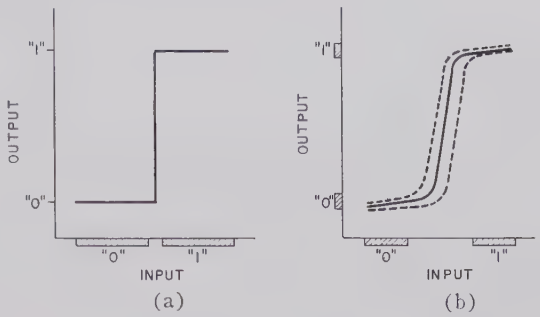


Fig. 1.—Signal standardization; illustrative characteristic of: (a) An ideal network. (b) A group of nominally identical, practical networks.

the input-output relations of some physical quantities in a network (*e.g.*, voltage, current, etc.). The actual value of these quantities depends on specific circuit arrangements. Take the case of a digital cell driving three similar cells in parallel. Here the output voltage of the cell should be the same as the standard input voltage, while the output current should be three times the standard input current.

As digital signals are observed at a prescribed location and during a prescribed time interval, timing is also involved in signal standardization, through the use of an unconditional time clock or a signal-derived time clock.

Signal standardization to the full extent need not be employed, of course, at every step of logical operation. In many practical cases, considerable deterioration in signals through several logical stages can be tolerated. Partial standardization, such as that employed in diode logic circuits (which supply discriminative attenuation, but no gain) and emitter-follower logic circuits (which supply current gain, but no voltage gain), is extensively practiced.

B. Directivity of Control

One distinct feature of a digital information-handling system is that it employs a large number of networks interconnected in a complicated manner. A predominant requirement in such a system is to insure that the controlling networks dictate the behavior of the controlled networks so that no ambiguity may occur through any spurious interaction. Take the simple networks arrangement shown in Fig. 2 where elements A, B, C, combinatorially, in a prescribed logical manner, control the element K, which in turn controls elements X, Y, Z, conditionally according to other controls. The arrows in this logical block diagram specify control directivity, and such path and sense of control as is illustrated by the dotted arrows should be prevented.

One obvious way to achieve control directivity is to use *unidirectional coupling elements* between digital cells to allow signal flow in only one sense in each path. (While diodes can be used for digital signals in the form of dc levels of the same polarity, other forms of digital signals may require a more general unidirectional coupling device, such as an isolator in microwave cir-

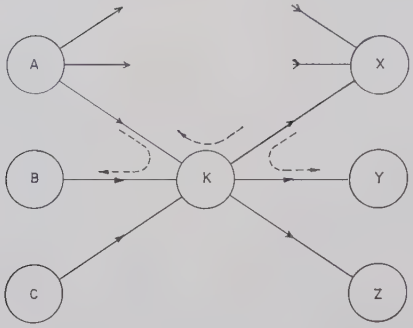


Fig. 2—Control directivity; desired and undesired path and direction of control.

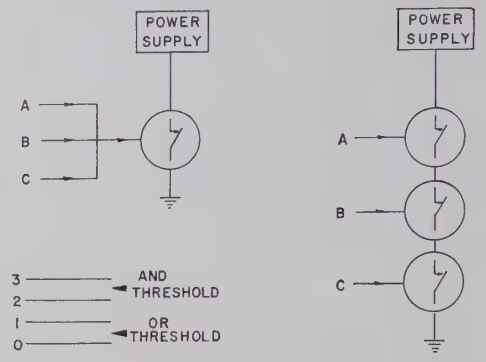


Fig. 3—Information manipulation. (a) Signal summation; threshold position decides AND or OR operation. (b) "State" summation in synthesized network.

cuits.) Blocking signal flow in any undesirable sense and any undesirable path is a sufficient condition to provide directivity. However, more generally, one must realize that spurious signal flow may be tolerated as long as the spurious signal does not lead to false control.

Separation of input and output signals in a digital-gain element is a first step toward providing control directivity. In 4-terminal or 3-terminal devices this is readily achieved through separation in *space* (*i.e.*, signals appear at different terminals). In 2-terminal devices a separation in *time* or in *frequency* may be employed. These techniques will be treated in detail when various digital-gain elements are discussed.

Unilateralization and *isolation* are major factors in providing directivity. Digital-gain elements with unilateral properties and unidirectional coupling elements are highly desirable to accomplish this task, but are not utterly indispensable (as illustrated in later discussions). Isolation of digital cells where interaction is undesirable can also be achieved by using linear attenuators as coupling elements. Referring to the example shown in Fig. 2, if a signal S is attenuated to S/n in traveling the path K to X or K to Y, a signal traveling the complete path X to K to Y will be attenuated to S/n^2, which is too small to cause any false control. Loss of signal through the attenuators must be compensated for by gain in the network, illustrating the concept that digital gain plays a significant part in providing isolation.

While the purpose of a digital network is information manipulation, the primary considerations in the physical operation of digital networks are signal standardization and control directivity. One may well take the view that most actual circuit design is done to fulfill these two prerequisites. In developing new components and circuit techniques these requirements can well serve as a useful guide.

C. Information Manipulation

Ways and means of executing information manipulation vary for various types of devices and circuits. Nevertheless, one can see that the physical operations fall in several general categories:

1) *Combinatorial Logic by Signal Summation:* As illustrated in Fig. 3(a), several input signals are summed and the resultant value is compared against a predetermined threshold value (of a device, with external bias if desired). Depending on this inequality the digital network goes to its "0" or "1" state, delivering a standardized "0" or "1" signal accordingly. The summation process can be linear, or weighted, or nonlinear, and it is performed in analog manner. However, one need not feel embarassed about allowing some analog operation in a digital system, since any digital signal in reality is quantitative in nature. The pertinent consideration is that signal standardization be practiced before ambiguity sets in.

2) *Combinatorial Logic by "State" Summation:* Fig. 3(b) shows the example of a network composed of digital elements in series (or in parallel, or in other series-parallel combinations), where each digital element is controlled by an input signal and assumes its "0" or "1" state accordingly. As a whole the network exhibits a distinct "0" or "1" state according to some prescribed logic combination of the states of the individual elements, delivering an output "0" or "1" signal accordingly.

The distinction between the two modes of physical realization of combinatorial logical functions, although apparent in circuit arrangements, should not be taken too seriously. In many practical cases, a mixture of the two modes, to suit specific device and circuit needs, is used.

3) *Negation* and *delay* are important operations not included in the above discussion. Physical realization of *negation* (*i.e.*, converting a "0" signal to a "1" signal, and vice versa) depends heavily on the nature of the signal and is executed by a specific device which provides this conversion. For signals whose "0" and "1" representations are mirror images of each other (such as a pulse of either positive or negative polarity, or a sinusoid of either 0° or 180° phase), a simple phase inversion can perform the *negation* operation. In the case of the carrier phase-script signal, a delay of half a cycle is sufficient. For signals in the form of dc levels a device with the "inversion" property (such as a tube or a transistor) is needed.

Delay can be physically realized by allowing passive delay (with a delay line, or other time-constant network), or by storage with time-clocked interrogation arrangements. A discussion of storage in relation to sequential operation is not included in this paper.

In recognizing the basic physical requirements one should keep in mind that many of the component features (such as device tolerance, digital gain, isolation, dissipation, and speed) are interrelated; and the trading of one for the other to reach a happy compromise is the fine art of engineering. In the following sections, where specific components will be discussed, this point will be illustrated in more detail.

III. An Illustrative Example

Digital networks employing parametric phase-locked oscillators (PLO) are used here to illustrate some of the generalized concepts described in the previous discussion. The PLO, being original and unusual (in the sense that it lacks many of the properties of more widely used digital devices and yet performs all the digital functions in a somewhat nonconventional manner), helps to bring out some of the pertinent points in the general discussion most revealingly.[1,2]

When energized by a pump of frequency $2f$, the PLO builds up an oscillation of frequency f in either 0° phase or 180° phase (with respect to a reference phase standard maintained by the pump, which also serves as a timing clock) as shown in Fig. 4. In its steady state, this oscillation is "locked" in phase, and maintains a constant amplitude. When energized in the absence of an input signal, oscillation of either phase may build up in equal probability. Control is executed by steering the oscillation to one of the two standard phases by an input signal applied prior to, or at the beginning of, the pump energization. The slightest input signal of roughly the 0° phase or the 180° phase is capable of steering oscillation build-up in the respective phase. When the oscillation is building up, it is also "pulled" to its respective standard phase angle. Sustained oscillation amplitude is determined by circuit parameters (primarily by the nonlinear characteristics of the variable reactance element, and losses in the circuit) and is not sensitive to reasonable variation in pump or input-signal amplitudes. Note that in the PLO system the phase (which represents information) is discriminatively standardized while the amplitude is indiscriminatively standardized. The digital gain of the PLO is large (fan-out greater than ten is readily obtained); but switching speed (number of cycles for oscillation to build up to steady-state amplitude) is greatly affected by input-signal amplitude and loading, illustrating the gain-speed relation involved in many digital devices.

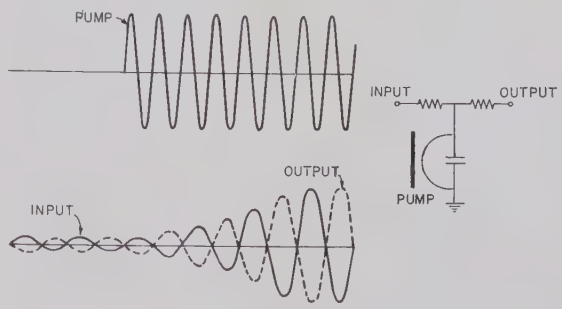


Fig. 4—Parametric phase-locked oscillator.

Control directivity is a major concern in PLO operation. Since PLO is a 2-terminal device, operating normally with one terminal grounded, the input and output signals share the same terminal. Control directivity is achieved through separation (of its input and output signals) in *time* rather than separation in *space* (as in 3- or 4-terminal devices). This operation is possible through a unique property of the PLO (and other devices of this class) that it is responsive to an input signal only during a short time interval (when the pump is first applied). Once the oscillation has built up enough in amplitude, its phase is locked by the pump, and signals of many times normal amplitude cannot change the phase information. As illustrated in Fig. 5, where information flow in the order P_1, P_2, P_3 is intended, the pumps are turned on and off in the sequence I-II-III with overlap in time of two pumps only. Now consider the time interval when pump II is ON and pumps I and III are OFF; signals do flow from P_2 to both P_1 and P_3 (both idle). In the next moment pump III is turned ON to cause P_3 to receive and to respond to the information from P_2, while P_1 is still idle. During this time there is no danger of P_2 being affected by P_3 (or other elements served by pump III) because the state of P_2 is already locked in by the pump and is immune to input signals. Control directivity achieved in this manner with 2-terminal devices in linear coupling is indeed an interesting illustration of the circuit possibilities in digital networks.

Signal manipulation in the "signal summation" manner is illustrated in Fig. 6. Linear summation of sinusoids of the same frequency, equal amplitude, and two phases 180° apart results in a sinusoid in one or the other phase. The sum of inputs A and B is compared against a reference R, and the resultant decides the state of the element K. The PLO may be considered to be a device with a natural and perfect threshold (in that it is perfectly symmetrical with respect to 0° and 180° phase oscillation); and logic operation with weighted inputs, nonlinear summation, and a large number of fan-ins, is theoretically possible, and has

[1] E. Goto, "The parametron, a digital computing element which utilizes parametric oscillation," Proc. IRE, vol. 47, pp. 1304–1316; August, 1959.
J. von Neumann, "Nonlinear Capacitance or Inductance Switching, Amplifying, and Memory Organs," U. S. Patent No. 2,815,488; December, 1957.

[2] L. S. Onyshkevych, W. F. Kosonocky, and A. W. Lo, "Parametric phase-locked oscillator—characteristics and applications to digital systems," IRE Trans. on Electronic Computers, vol. EC-8, pp. 277–286; September, 1959.

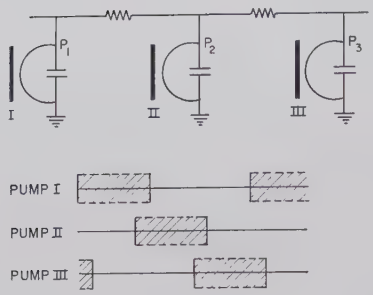


Fig. 5—Sequential pump energization provides control directivity in PLO operation.

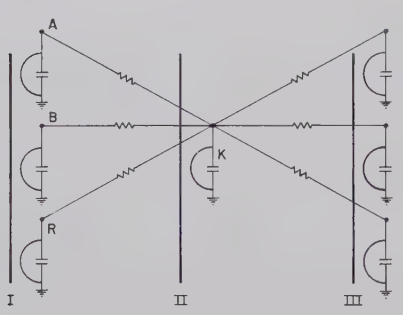


Fig. 6—Majority logic.

been suggested. In practice, the major considerations are, however, the uniformity (in amplitude) of input signals (including the reference) and the speed of operation (which is sensitive to input signal strength). Being a carrier phase-script signal, *inversion* can be readily achieved by a transformer, or by delaying the signal a half a cycle time.

Isolation between digital cells sharing the same pump is achieved here with the use of a linear, bilateral attenuator (resistive or reactive) as a coupling element. This attenuation of signal strength is, of course, made up by the digital gain of the PLO.

IV. A Listing of Digital Gain Elements

Since digital gain plays such an important role in signal standardization and control directivity, one may take the view that digital circuit techniques are developed around digital-gain elements. The history of the development of digital-computer circuits seems to support this viewpoint. A survey of the large number of devices capable of providing digital gain reveals that while differences in structure, behavior, and mode of operation are apparent, these devices can, nevertheless, be cataloged in a systematic manner. With a systematic approach one can hope to bring out some operation principles generally shared by digital-gain elements in each group, and thus offer some guidance for the search for and development of new devices.

Taking the view that the "state" of an electronic element is measured by its reaction toward some electrical energizing source, one can generalize to say that impedance is the physical parameter which represents the state of a digital element. To classify digital-gain elements according to physical operation is therefore a classification of the modes of executing impedance control in the element. A listing of digital-gain elements compiled in this manner is shown in Table I.

TABLE I
Digital-Gain Elements

- A. Continuous Control
  - 1) 4-terminal resistance-controlled elements
    - a) Electromechanical relays
    - b) Cryoelectric devices
    - c) Optoelectric devices
  - 2) 3-terminal resistance-controlled elements
    - a) Tubes
    - b) Transistors
  - 3) 2-terminal reactance-controlled elements
    - a) Nonlinear inductors
    - b) Nonlinear capacitors
- B. Stored Control
  - 1) Carrier-storage devices
  - 2) Ferroelectric devices
  - 3) Ferromagnetic devices
- C. Regenerative Control
  - 1) Parametric phase-locked oscillators
  - 2) Negative-resistance diodes

A. Continuous-Impedance Control

In this class of elements the device impedance is dictated by the input signal present at that instant. The group is subdivided into:

1) *Four-Terminal Resistance-Controlled Elements:* Devices in this group are illustrated in Fig. 7(a), and the diagrams are self-explanatory. An input signal below or above a threshold range causes the resistance between two specified terminals of the device to stay in a distinctly high or low value. With an energizing source (not shown) a "0" or "1" output signal is derived (as current in series with the element, or voltage across the element) according to the resistive state of the device. Some features of this group of elements are noteworthy: a) a sharp and quite well-defined threshold, b) well-defined and very distinct "0" and "1" states, c) excellent unilateral property between input and output, d) large digital gain, and e) the fact that the four terminals permit flexibility in logical connection.[3–5] It is to be noted that the excellent unilateral property is a result of using more than one form of physical energy in its operation. (For example, in the optoelectric case the EL material produces light from electric energy but is insensitive to light; while the PC material is electrically sensitive to light but does not produce light.) This may be a point of interest in the search for new digital elements. All these devices, in the present state of the art, suffer in switching speed; but there seems to be no basic physical theory dictating such limitations. A significant

[3] This class of 4-terminal devices can be extended to include multiterminal devices where a number of output (and/or input) circuits isolated from each other can be fabricated in one device. The electromechanical relay with multiple windings and multiple contacts is a practical example.

[4] D. A. Buck, "The cryotron—a superconductive computer component," Proc. IRE, vol. 44, p. 482–493; April, 1956.

[5] E. E. Loebner, "Opto-electronic devices and networks," Proc. IRE, vol. 43, pp. 1897–1906; December, 1955.

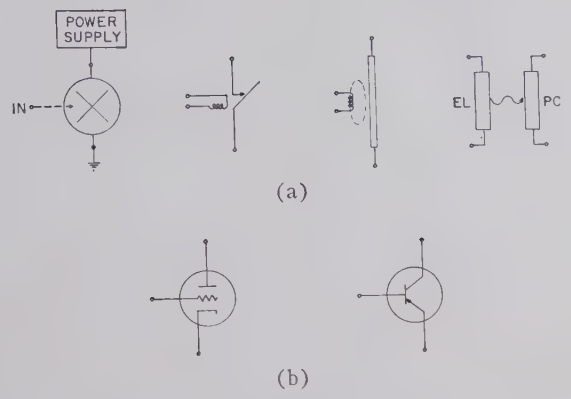


Fig. 7—Resistance-controlled digital-gain elements. (a) 4-terminal elements, electro-mechanical relay, cyroelectric element, opto-electric element. (b) 3-terminal elements, tubes and transistors.

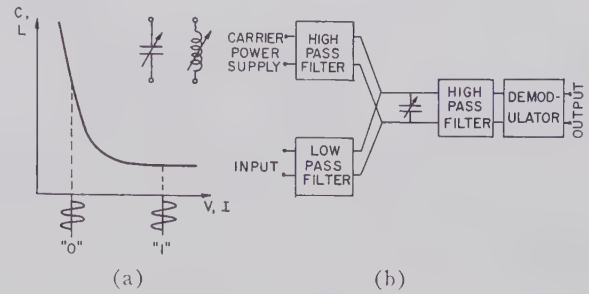


Fig. 8—Reactance controlled digital-gain elements. (a) Operational characteristic. (b) Filters provide isolation.

fact here is that both cryoelectric and optoelectric elements employ bulk effect, homogeneous material. The seemingly complicated 4-terminal structure actually can be viewed as two simple 2-terminal elements placed in close (but not very critical) proximity. This property may lead to extremely simple fabrication techniques for producing very large numbers of nominally identical elements by an automatic manufacturing process.

2) *Three-Terminal Resistance-Controlled Elements:* These digital-gain elements, including various forms of tubes and transistors, known as "active elements" in linear network terms, are the "work horses" in present digital systems. The operation of these devices is well known and the mechanism of achieving gain is also well understood.[6] Their success, one may say, hinges on their excellent gain and unilateral properties in fulfilling the signal standardization and control directivity requirements. These devices, comparatively speaking, have a rather complicated structure of critical geometry. Improvements in uniformity and operation speed through better manufacturing techniques have reached a high plateau.

3) *Two-Terminal Reactance-Controlled Elements:* This class includes the nonlinear reactive elements operated with a *carrier* energizing source. (While devices with hysteresis characteristics often serve well as nonlinear elements, devices that obtain digital gain through their remanence or storage property will be treated as another class.) As illustrated in Fig. 8, a "0" or "1" input signal places the operating point in a low- or high-reactance region, respectively. The output signal is derived through a carrier energizing source acting in that region. Gain is obtained through the fact that input and output signals, being different in frequency, see different reactances in the element. The operation is characterized by a large excursion of the operating point produced by the input signal at low frequency in contrast to a small excursion at high frequency produced by the energizing source. This relation between gain and the ratio of carrier frequency to signal frequency is another example of the gain-speed relationship.

While there is no separation in *space* (for input and output signals) in these 2-terminal elements, the difference of input and immediate output frequencies offers separation (and thus unilateralization) in *frequency*. As illustrated in Fig. 8, the high-pass and low-pass filters provide control directivity. Although this group of elements has not been extensively used in present digital systems, except in some cases of ferroresonant circuits,[7] their use at microwave frequencies, where filtering and other circuit techniques actually become simple, seems to be one of the feasible means of achieving very high-speed digital computation.[8]

B. Stored-Impedance Control

This category includes elements whose digalit state is stored and remains in the element after the termination of the input signal. Initially the element is in its *normal* state. An input "1" shifts the element to its *abnormal* state. Subsequently an energizing source is applied, unconditionally with respect to the stored information, to return the element to its normal state. Digital gain is obtained by the fact that a small input signal is capable of a change of state. A large energizing source can be applied to sense whether a change has occurred, and to derive a large output signal accordingly.

The operation of a storage diode is illustrated in Fig. 9.[9] An input "1" stores minority carriers in the diode in forward conduction (thus small input energy). With the presence of stored excess carriers, the diode shows a low resistance to an energizing pulse applied in the reverse direction (of the diode) until the excess carriers are swept out of the diode. In the absence of stored carriers the energizing pulse sees a high resistance. One may say that the gain is a result of the fact that the carriers are put into the device at low energy and taken

[6] E. O. Johnson and A. Rose, "Simple general analysis of amplifier devices with emitter control and collector functions," PROC. IRE, vol. 47, pp. 407–418; March, 1959

[7] C. B. Newport and D. A. Bell, "Ferroresonant circuits for digital computers," *J. Brit. IRE*, vol. 17, pp. 619–630; November, 1957.

[8] W. Eckhardt and F. Sterzer, "A Modulation-Demodulation Scheme for Ultrahigh-Speed Computing and Wideband Amplification," presented at Internatl. Solid-State Circuits Conf., Philadelphia, Pa.; February 10–12, 1960.

[9] A. W. Holt, "Diode amplifier," *Radio Electronics Engrg.*, vol, 24, pp. 18–19; January, 1955.

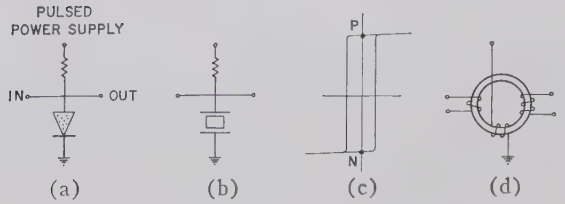


Fig. 9—Digital-gain elements employing storage. (a) Storage diode. (b) Ferroelectric element. (c) Hysteresis loop. (d) Ferromagnetic element.

out at higher energy, the same carriers being involved in the operation. This is in contrast to the case of tubes and transistors where some charges are placed inside the device to allow *other* charges (in larger quantity) to be taken out at higher energy.[6]

The storage in the diode case is volatile and requires that pulse energization be applied immediately after the input signal. In the case of ferroelectric and ferromagnetic elements the storage is nonvolatile.[10,11] As illustrated in Fig. 9(c), the element displays a normal (N) state and an abnormal (P) state with static storage (no holding power) and sharp and well-defined thresholds. The square-loop magnetic elements are used extensively in some digital systems; and their operations serve well to bring out some pertinent points in the operation of digital elements in general.

Consider an isolated square-loop core with single-turn input and output windings (Fig. 10). With the load to the core disconnected, an input signal of voltage-time product equal to $\int edt = \Phi$ and of current greater than a critical value I_c is sufficient to switch the core from its N state to its P state. Now if an energizing pulse is applied, with the load connected to the output winding, the energized source delivers to the load the same voltage-time product (limited by the amount of flux set in the core), and a current *not* limited by the core but only by the current capacity of the source. In this way the element supplies an output current greater than its input current (not at the same time, of course). In magnetic circuits voltage and current can be readily traded for each other through transformer action. Thus, both flux gain (a measure of $\int edt$) and current gain can be easily achieved simultaneously by the use of a transformer, or simply by having more turns in the output winding than in the input winding. In this manner one core can drive several cores in series (with flux gain) or in parallel (with current gain) or in other combinations.

The difficulty in magnetic circuits lies in the problem of isolation. Although a core can have a number of windings, the interaction between these windings, when a flux change occurs, makes the device behave like a

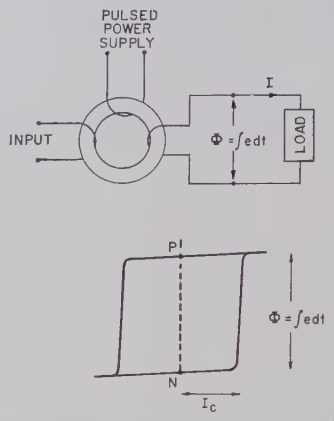


Fig. 10—Operation of square-loop core to obtain digital gain.

2-terminal device. When cores are interconnected with each other it is imperative that certain paths be blocked to prevent false control and to prevent undesired loading. This is illustrated in Fig. 11. When it is intended to transfer information from core 2 to core 3 (through path "b"), path "a" and path "c" are blocked and/or attenuated to insure that cores 1 and 4 are not disturbed by the signals, and that cores 2 and 3 are not loaded down.[12]

To obtain gain in the above mentioned manner, the input and output signals cannot occur at the same time; this requires a time-sequence energization system. The use of two or three clocks does not in itself offer control directivity here as in the PLO, since the cores do not have the property of responding to an input signal only in a specified time interval.

Much sophistication has gone into developing circuit techniques to provide control directivity in magnetic logic circuits without the use of unidirectional coupling elements.[13–15] To illustrate some interesting points let us consider a case of operating transfluxors interconnected with linear resistors, as shown in Fig. 12. Transfluxor B begins in the "clear" state [Fig. 12(b)] after an "advance" pulse is applied. A "1" input signal produces in the transfluxor flux transfer between legs 1 and 3 [Fig. 12(c)]. Since this flux change is not linked to the output winding, no output signal is produced, nor is the transfluxor loaded down during this switching. The next step is a priming operation executed by sending a current through the two small holes in the sense that it causes flux transfer between legs 1 and 2, and between legs 3 and 4 [Fig. 12(d)]. Now these flux changes do link with input and output windings, and special consideration is required.

[10] J. R. Anderson, "Ferroelectric storage elements for digital computers and switching systems," *Elec. Engrg.*, vol. 71, pp. 916–922; October, 1952.

[11] J. A. Rajchman, "Magnetic switching," *Proc. Western Joint Computer Conf.*, Los Angeles, Calif., May 6–8, 1958; pp. 107–116.

[12] A. Wang and W. D. Woo, "Static magnetic storage and delay line," *J. Appl. Phys.*, vol. 21, pp. 49–54; January, 1950.

[13] J. A. Rajchman and A. W. Lo, "The transfluxor," Proc. IRE, vol. 44, pp. 321–332; March, 1956.

[14] H. D. Crane, "A high-speed logic system using magnetic elements and connecting wire only," Proc. IRE, vol. 47, pp. 63–73; January, 1959.

[15] U. F. Gianola, "Integrated magnetic circuits for synchronous sequential logic machines," *Bell. Sys. Tech. J.*, vol. 39, pp. 295–332; March, 1960.

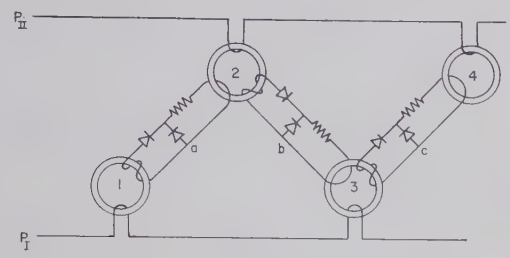


Fig. 11—A core-diode shift register.

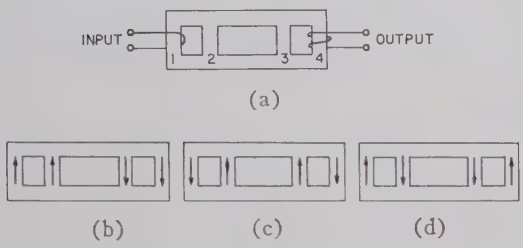


Fig. 12—Operation of a transfluxor when connected to other transfluxors with linear, resistive coupling elements.

The output-voltage waveform of a core being switched over depends heavily on the switch-over time, although the quantity $\int edt$ is a constant. As shown in Fig. 13, the output voltage of a core in fast switching shows a peak value considerably higher than that of the core in slow switching, particularly for slow magnetic-core material. The result is that with fast switching (by a large drive current) the core delivers enough voltage to cause a current greater than the threshold current I_c for a length of time required to switch the next core; and with slow switching (by a small drive current) the output current is below I_c and the second core is not affected. One can take the viewpoint that in slow switching the coupling resistance absorbs all the $\int edt$ quantity from the output winding, while in fast switching part of this quantity is delivered to the next core, and the rest is absorbed by the coupling resistance.[15–17]

Returning to the transfluxor circuit in Fig. 12, one sees that priming of the transfluxor by slow switching with a well-chosen prime current (its amplitude has to be limited anyway to avoid spurious setting in case the transfluxor is in its "0" state) can be executed without affecting other transfluxors connected to the input and output windings. The next operation is the application of a powerful "advance" pulse in the large hole, which causes a fast flux transfer between legs 2 and 4 and sends a signal in the output winding to act on the next transfluxor, while leaving the input winding not disturbed. This advance pulse returns the transfluxor to its cleared state, ready for the next logic operation. This case of trading speed for isolation, not known in conventional circuits, illustrates the flexibility of digital network techniques and the intimate relation between the physical behavior of a device and the circuit considerations.

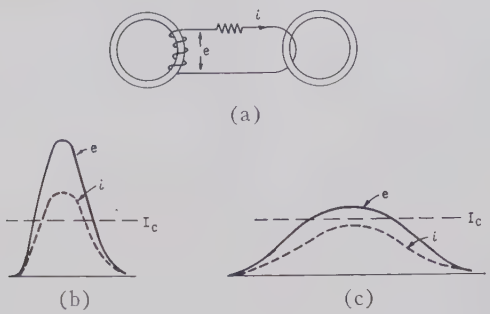


Fig. 13—Square-loop core output waveforms for: (a) Fast switching. (b) Slow switching.

C. *Regenerative-Impedance Control*

With proper *external* feedback arrangements, a gain element can be operated in regenerative manner. Some devices, however, have internal regenerative properties which make them particularly suitable for digital operation. The outstanding members of this category are the parametric phase-locked oscillators[18] and the negative resistance diodes including the tunnel diodes.[19,20] The PLO has been treated earlier; the operation of the tunnel diode in several interesting modes will be treated here.[21–23]

The tunnel-diode static characteristic exhibits two distinct "constant" voltage regions (representing the "0" and "1" states) separated by a negative-resistance region. Digital gain is obtained by virtue of the negative resistance and can be achieved by several different modes of operation.

The *polarity-locking* mode of operation of the tunnel diode as a digital-gain element is shown in Fig. 14. Here an input signal of one polarity or the other causes regeneration (when energized by a balanced voltage pulse source) in one diode or the other, respectively, to form an output signal of the same polarity as the input signal, for the duration of the energizing pulse. The circuit is immune to input signals except for a short duration at the very beginning of the energizing pulse, permit-

[15] G. R. Briggs, "A Magnetic Core Gate and Its Application in a Stepping Register," Digital Computer Lab., Mass. Inst. Tech., Engrg. Rept. E-475; October, 1952.

[17] L. A. Russell, "Diodeless magnetic core logical circuits," 1957 IRE Natl. Convention Record, pt. 4, pp. 106–114.

[18] Calling the operation of the PLO a form of impedance control is indeed a far stretch of imagination, even dragging in the negative resistance concept. It is, nevertheless, included here to complete the list of digital gain elements.

[19] L. Esaki, "New phenomenon in narrow Ge *p-n* junctions," *Phys. Rev.*, vol. 109, pp. 603; January, 1958.

[20] H. S. Sommers, Jr., "Tunnel diodes as high-frequency devices," Proc. IRE, vol. 47, pp. 1201–1206; July, 1959.

[21] M. H. Lewin, "Negative-resistance elements as digital computer components," *Proc. Eastern Joint Computer Conf.*, Boston, Mass., December 1–3, 1959; pp. 15–27.

[22] M. H. Lewin, A. G. Samusenko, and A. W. Lo, "The Tunnel Diode as a Logic Element," presented at Internatl. Solid-State Circuits Conf., Philadelphia, Pa., February 10–12, 1960.

[23] E. Goto, *et al.*, "Esaki diode and high speed logical circuits," IRE Trans. on Electronic Computers, vol. EC-9, pp. 25–29; March, 1960.

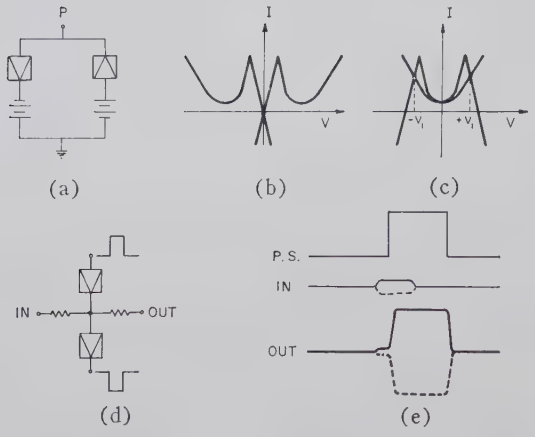


Fig. 14—Operation of a tunnel diode in the "polarity-locking" mode. (a) Illustrative dc circuit where (b) output terminal P is at ground potential when circuit is not energized, and (c) terminal P is at either $+V$ or $-V$ when circuit is energized. (d) Circuit is energized by balanced voltage pulse source. (e) Output waveforms showing output polarity following that of input signal.

ting the same type of circuit and logic organizations used in the PLO system.[1] There are, however, some inherent differences between the two devices. While the PLO has a natural and perfect threshold (to the "0" and "1" phase-script signals), the tunnel-diode circuit is balanced artificially and thus requires a minimum input signal to overcome any asymmetry in the circuit. Matching of a pair of diodes is needed, while the tolerance between pairs is not so critical. The PLO uses the carrier type of operation, and the information-handling rate is usually a fraction of the pump frequency; on the other hand, the tunnel diode is pulse-operated, and the information-handling rate is close to the energizing pulse (which can well be a biased sinusoid) frequency.

Another mode, termed *bistable operation with unconditional reset*, is illustrated in Fig. 15. Quiescently, the diode is in its "0" state, at an operating point P just below the current peak. The diode at this low-voltage state delivers little current to its load. A small input signal (I_1) is capable of triggering the diode to its "1" (high-voltage) state. Without the load drawing any current the new operating point would be point Q; but with loads drawing a total current, say, $3I_1$, the operating point moves down to point S. The maximum allowable load current (for the diode to stay in the "1" state) is limited by the fact that point S must be kept above the knee of the characteristic. The current gain of the device theoretically can be made extremely large by biasing the operating point P very close to the peak current point. The limit is a practical one, dictated by the uniformity of the diodes (especially in peak current) and the other components in a system. Furthermore, the switching speed of the diode is greatly affected by the amount of over-drive (*i.e.*, net current in excess of peak current). These examples illustrate the gain-uniformity relation and the speed-gain relation mentioned earlier.

The gain in such an operation, however, is a "one-way" gain; that is, gain is realized in switching from "0" to "1" state, but not vice versa. In fact, once in the "1" state the device is immune to any normal signal. Thus after each logic operation the device has to be reset to its "0" state (by a reset pulse or termination of the dc source) before another operation can be performed. This mode of operation, characterized by asynchronous "1" propagation (the "0" signal or state does not propagate) and reset, is nevertheless useful in certain applications.

The *monostable* mode of operation (Fig. 16) is in many ways similar to the bistable operation, except that the "1" output lasts only for a fixed time duration, and the circuit resets itself to its "0" state (with a considerable recovery time) following a "1" output. Synchronous operation is required in this and other monostable operations.

Unlike the polarity-locking mode of tunnel-diode operation and PLO operation, the tunnel diodes in the bistable and monostable operation modes are not immune to input "1" signals when they are in the "0" state. The energizing source does not "lock-in" in the "0" state; and the use of 2-or-3-clock energization alone cannot provide control directivity.[21] Usually unidirectional coupling elements are employed for this purpose.

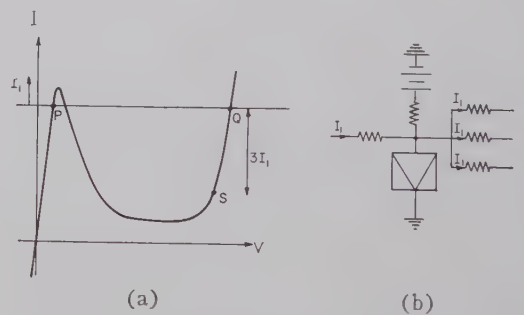


Fig. 15—Operation of tunnel diode in the "bistable-with-reset" mode.

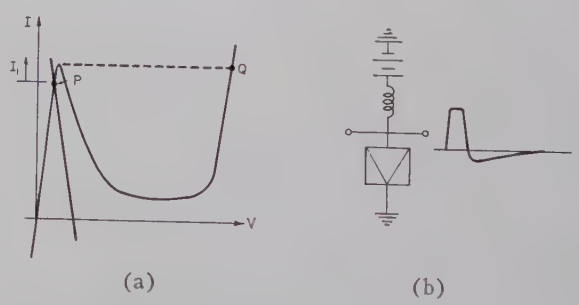


Fig. 16—Operation of tunnel diode in the "monostable" mode, showing output waveform.

V. Modes of Gating

Instead of classifying digital-gain elements according to their physical operation, one can also classify the circuit operation of these elements according to their gating modes. As illustrated in Fig. 17, the GO, NO GO gate allows energy from an energizing source to go through or not go through the gate according to an input "0" or "1" signal. Output signal is derived across the gate or in series with the gate as desired. In the

second class, the GO ONE WAY, GO OTHER WAY gates, the source energy always goes through the gate, but in one of two ways according to the input signal. This is illustrated by the polarity-locking tunnel-diode circuit and the PLO circuit, the latter being a more sophisticated mode, where a transformation of frequency is involved. The third class of gates, the GO NOW, GO LATER gates, operates in the manner that if an energizing source is applied for a long enough duration the energy will go through the gate; but an input signal decides the *delay* time. This operation is illustrated by one mode of operation of the PLO (Fig. 18).[24] The

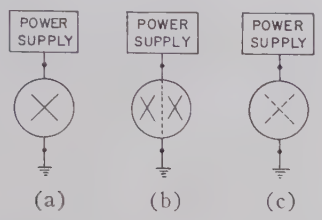


Fig. 17—Three modes of gating. (a) GO, NO GO gate. (b) GO ONE WAY, GO OTHER WAY gate. (c) GO NOW, GO LATER gate.

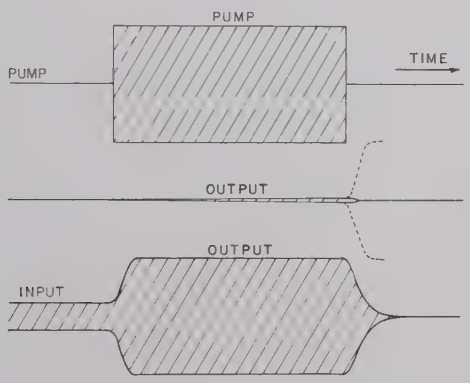


Fig. 18—Operation of the PLO in the "delay-controlled" mode.

switching time of a PLO is characterized by a distinct delay time and a short rise time. As shown in the figure, with a normal input signal oscillation builds up with little delay when energized by the pump. In the absence of a substantial input signal the delay time is long; now, if the pump is terminated in time, no substantial oscillation is produced. This effect was also observed in the operation of impact ionization devices.[25] While the gates of the first class are the only ones extensively used in present digital systems, the other two classes of gates cannot be overlooked in the search for and development of new devices and circuits.

VI. Some Further Thoughts

1) The formation of a general theory to cover digital networks of all varieties seems to be a formidable task at present. Immediate efforts may be directed toward identifying the pertinent physical and functional characteristics and requirements common to many, if not all, digital networks. Concepts and terminology borrowed from linear network theory soon become inadequate in representing situations in digital networks; new and well-defined terms need to be created. (For example, how do we define gain in digital networks? Does the gain-bandwidth figure of merit have a counterpart in digital systems? Can we compare the merit of a transistor to that of a ferrite core in certain digital operations?)

2) Treating devices and circuits as two separate subjects is no longer practical. With the advent of electronically active solids, studies of materials, devices, and circuits all merge into one. What physical phenomena occur in which materials? What devices can be made out of these materials? What modes of operation of these devices in what circuit arrangements can perform the desired system functions? In answering these questions the need for a coordinated approach is obvious.

3) The size and complexity of modern digital systems point to the inevitable future of integrated electronics (in all degrees from "micromodule" to "molecular engineering"). Development of fabrication techniques to produce a large number of components interconnected in a complicated manner is as important as development of components and circuits. The availability and adaptability of certain fabrication techniques will exert great influence on the choice of components and circuits in future machines.

VII. Acknowledgment

The author is indebted to his many colleagues for invaluable discussions and comments.

[24] P. Schnitzler, "A note on the delay in tunnel diode switching," to be published.

[25] M. C. Steele, L. Pensak, and R. D. Gold "Pulse amplification using impact ionization in germanium," Proc. IRE, vol. 47, pp. 1109–1117; June, 1959.

UNIVAC-LARC High-Speed Circuitry: Case History in Circuit Optimization*

N. S. PRYWES†, MEMBER, IRE, H. LUKOFF‡, SENIOR MEMBER, IRE, AND J. SCHWARZ‡, MEMBER, IRE

Summary—**In recent years, the design of computer circuits has become a thorough and complex job. Factors such as logical design, behavior of components, manufacturing techniques, and life tests are playing an even more important role in the design of an efficient circuit.**

This paper will discuss how circuit optimization techniques and use of the UNIVAC® I computing system aided in reducing cost and avoiding many of the pitfalls in the design and production of efficient high-speed circuitry for UNIVAC-LARC.

Introduction

BECAUSE computing systems have become so large and complex, computer circuit design has undergone a major change in the past five years. Today's designer of computer circuits must not only be proficient in electronic circuitry, but he must also understand logical design, the physics of the components with which he is working, and the methods used in producing the components. He must proceed more methodically than formerly and use new tools—such as computers—in designing new circuits. This paper deals with the case history of such a design, illustrating the demands on the designer of computer circuits.

As a fitting example of circuit design, we have chosen some design features of the UNIVAC-LARC computing system. The LARC system was and still is a very ambitious undertaking [1]–[4]. It is a true multiple large computing system consisting of two computing units, one processor, up to 13 high-speed synchronizers for input-output and mass-storage devices, and up to 39 random-access memory units, all operating independently and communicating with each other.

"Optimization" in computers is that mechanization of equipment which results in maximum performance at minimum cost. The problem, fortunately, is greatly simplified by the availability of performance specifications of the equipment to be designed. In the example outlined in this paper, optimization is used for reduction of cost only.

Circuit Optimization

In mechanizing the information processing, the designer must choose one scheme from a multiplicity of schemes. What follows is a process for evaluation of the various schemes and a singling out of one scheme and the design for obtaining the best performance in the computer. Even with a specific example, this scheme has general applications and has been utilized successfully on other projects at Remington Rand Univac.

The process of optimization usually becomes more rigorous as the number of alternative designs is reduced. For instance, in the evaluation of many possible circuitry configurations, comparison of the configurations must be partially qualitative. Because the amount of knowledge involved consists of the entire field of computer engineering, it would be very laborious to organize all the information so that it is all brought into account at the same time. The partially qualitative approach at this stage of optimization, however, does not detract from the validity of the results. It is not claimed that an actual minimum cost resulted from the study. As a matter of fact, with benefit of hindsight, we can see further possible improvements. However, we feel that, as a result of the study, we are quite close to the ideal objective. Cost of the study should be included in the total cost. As the optimization is narrowed to one circuit, more rigorous processes can be followed.

This paper will attempt to answer three questions:

1) Optimization for What? The answer is that the equipment is designed to conform to specifications at minimal equipment cost. We will show that cost can be reduced through:

a) Increase in speed;
b) Reduction in number of components;
c) Relaxation of component specification and tolerances;
d) Relaxation of specification and tolerances for power supplies and clock;
e) Increase in reliability, ease in debugging and servicing, marginal checks, and so forth, which reduce the cost of test and maintenance; and
f) Simplification of packaging and buses.

2) What Components to Optimize? The answer can be obtained only through a program of component evaluation.

3) How to Synthesize These Components into a System? The answer to this question will be arrived at by:

a) Evaluation of various types of circuitry;
b) Optimization of the chosen circuit;
c) Choice of a timing scheme.

In the case of LARC system, performance specifications indicated that very complex logic was to be performed. Therefore, the most important component of

* Received by the PGEC, March 2, 1961.
† Moore School of Elec. Engrg., University of Pennsylvania, Philadelphia, Pa.; consultant to Remington Rand Univac.
‡ Remington Rand Univac, Division of Sperry Rand Corp., Philadelphia, Pa.

the system is the logical gate. Since the number of logical-gate circuits exceeds the number of circuits used for other purposes in the system, the logical gate was the first circuit optimized. In the following text, the optimization of the circuit will be discussed in some detail. Similar procedures were followed with other circuits, some of which will be discussed toward the end of this paper.

The Derivation of Circuit Specifications

The first step in the process of optimizing the logical gate is to set design objectives for the delay per gate [the measuring technique of delays is illustrated in Fig. 9(a)], fan-in (number of inputs to a logical gate), and fan-out (number of gates to which an output may be connected). This objective should be obtained from the machine performance specifications. Examination of the latter is necessary to find which part of the specification will be the severest to comply with in terms of delay per gate, fan-in, and fan-out.

An examination of the LARC system specification indicated that the multiplication process represented the severest requirements. The specification called for the multiplication of two 11-decimal-digit numbers in 8 μsec. The multiplication process consists of three parts:

1) Generation of control signals and the transferring of operands to the proper registers.
2) Generation of multiples of the multiplicand.
3) The performance of 11 additions (one addition per multiplier digit).

A choice of a multiplication algorithm was made subject to the restrictions that floating-point and fixed-point multiplication and division be compatible with a minimum of special devices required for each operation over the basic hardware common to all. It was also desired to employ only one parallel adder to avoid the problem of providing switching currents required for multiple parallel operations. The considerable expense of checked parallel adders was an important consideration. Sixteen pulse times was chosen.

Addition time can be determined as the 11 additions should take the majority of the 8-μsec period. The multiplication was divided into 2 μsec for the first and second parts of the multiplication process; $5\frac{1}{2}$ μsec for the 11 additions (at the rate of 1 addition per $\frac{1}{2}$ μsec); and $\frac{1}{2}$ μsec for the final propagation of carry. Thus, the requirement for a $\frac{1}{2}$-μsec adder was obtained. The addition process was broken into three steps: 1) selecting input, complementing, gating, and so forth; 2) addition; and 3) unit add of carry.

The final step for obtaining the specification for the gate was the logical design of a number of adders, each having different numbers or levels of gating. The result of this logical design is shown by a graph in Fig. 1 [5]. Fig. 1 shows that as the number of levels of gating (the maximum number of gates through which information is propagated) increases from four to nine, the number of gates, number of diodes, and fan-out decrease rapidly. When the number of levels increases beyond nine, the improvement is far less. Therefore, the logical design using nine levels of gating was initially decided upon.

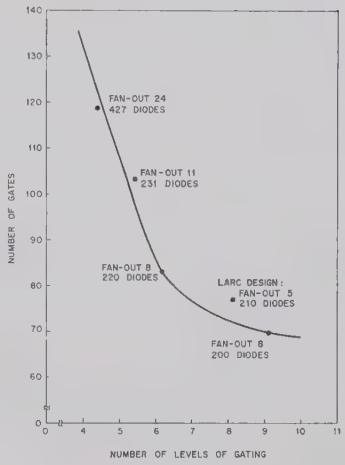


Fig. 1—Statistics of adder design.

In the process of designing the circuitry, we found that a fan-out of eight (corresponding to nine levels desired) was difficult to achieve. A compromise design (shown on Fig. 1) consisting of a fan-in of 13, 8 levels of gating, and a fan-out of 5 was attained, but one additional level of gating was necessary to amplify some signals. The final design actually uses a hybrid of the gate (gate C) that is discussed in this paper, and one other gate (gate E) which allows for an increase in fan-out at an expense in timing (equivalent to a delay of one level).

In conclusion, we found that we wanted information to propagate through nine levels of gating per $\frac{1}{2}$ μsec. We also anticipated an equivalent delay of approximately three levels of gating in pulseforming and restoring of information. This is equivalent to propagation through a total of 12 gates for a $\frac{1}{2}$-μsec addition period, which results in a maximum delay per gate of approximately 40 mμsec. Although the design called for a fan-out of 5, an effective fan-out of only 3 would be sufficient if mutual exclusion (load sharing) was employed in calculating the load on individual gates.

Evaluation of Components

The second step in the process of design was the evaluation and comparison of components which are suitable for computer circuitry. (This implies a cursory examination of circuitry in parallel with component evaluation.) The components evaluated included transistors, diodes, FERRACTOR® amplifiers, ferrite cores, resistors, capacitors, and so forth.

As an example of component evaluation, we have chosen the procedure followed in evaluating the transistor. Transistor parameters of primary importance are these:

1) Rise time (which is correlated with cutoff frequency). Rise time is measured for a change in collector current when applying a step in base current from zero to I_b optimum. [I_b optimum defined in (5).]
2) DC large signal beta current gain ($\beta = I_c/I_b$).
3) Breakdown voltage test.
4) Storage time. Storage time is measured as base current is changed from I_b optimum to $-I_b$ optimum, with a collector current during storage time of $4I_b$ optimum (Fig. 2).
5) Current level of optimum performance. Establish optimum current level by varying I_b or I_c until β reaches its maximum value. (Larger I_c results in greater speed and noise reduction.)
6) Cost. Cost estimates require the extrapolation into the future in considering the following:
   a) Mechanization of transistor manufacturers' facility.
   b) Estimate of market for a transistor type.
   c) Prospects for improvement in transistor characteristics by using new manufacturing techniques.

Such an investigation, conducted during the first half of 1956, resulted in the selection of Philco's surface-barrier transistor (similar to SB-100).

The evaluation of a large number of transistors by placing them in life and production tests requires many measurements. Generally, a single measurement is preferred to multiple measurements, a meter measurement is preferred to a scope measurement, and a scope amplitude measurement is preferred to a scope time measurement. Therefore, it was very important to design testers that could be operated, with a minimum of errors, by unskilled personnel.

Fig. 2 illustrates a storage-time tester which requires only a single meter measurement. The collector current during storage time is approximately constant. This current charged a capacitor whose voltage was therefore proportional to the storage time of the transistor. This voltage was then amplified and measured on a voltmeter which reads peak voltages. Similar techniques were followed in measurement of other parameters and components.

To attain measurements from components with extremely short response time, it was necessary to measure the response with special high-speed equipment. Such a case was a diode which had a recovery time in the order of a few millimicroseconds.

Fig. 3 shows a circuit diagram of the diode-reverse recovery tester. To simplify the measurement, the scope was later replaced by an integrator and a special peak-reading voltmeter. This resulted in reduced accuracy, but it was still satisfactory for large-scale production testing. Figs. 2 and 3 also demonstrate the care required in the design of testers.

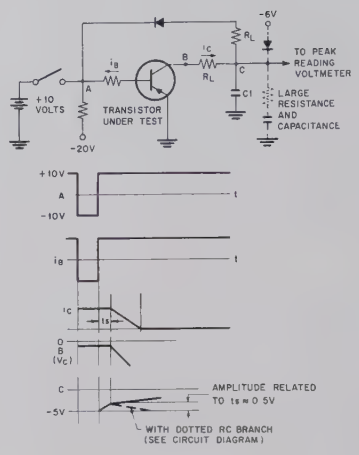


Fig. 2—Storage-time tester.

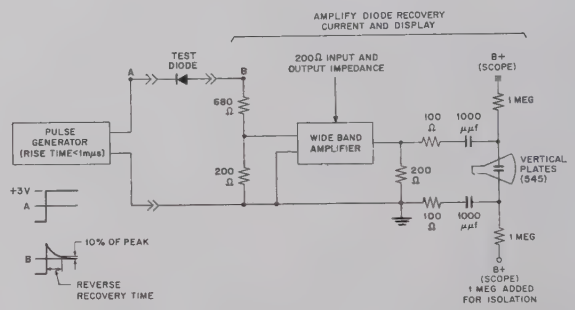


Fig. 3—High-speed reverse recovery-time tester.

In evaluating the life tests conducted, the use of computers was required to obtain significant results. Because of the long life expectancy of transistors, it was difficult to ascertain failure characteristics by life test over a true time scale [10]. In other words, we were not able to detect significant deterioration in transistor parameters over a life test of thousands of hours, but it was still extremely important to be able to make a prediction as to the life expectancy of the transistor.

An attempt was made to run accelerated life tests under elevated temperatures, severe temperature and humidity conditions, and vibration. The purpose of these tests was to produce a gradual deterioration in a reasonable time, and correlate the results of such studies with life test results which were performed under normal conditions. The data obtained from the accelerated life test were fed into the UNIVAC I system to determine pertinent statistics, moments, media, and so forth. Laws governing correlation between temperature and the age of transistors were also investigated. The deterioration of transistor breakdown voltage was found to be a function of both time and temperature of storage. By means of this function, we were able to extrapolate and estimate breakdown voltage deterioration at 25°C, based on similar data at elevated temperatures. The test resulted in a prediction of satisfactory transistor life of over 200,000 hours. Life expectancy due to a gradual decrease in the punch-through voltage is shown in Fig. 4 [11].

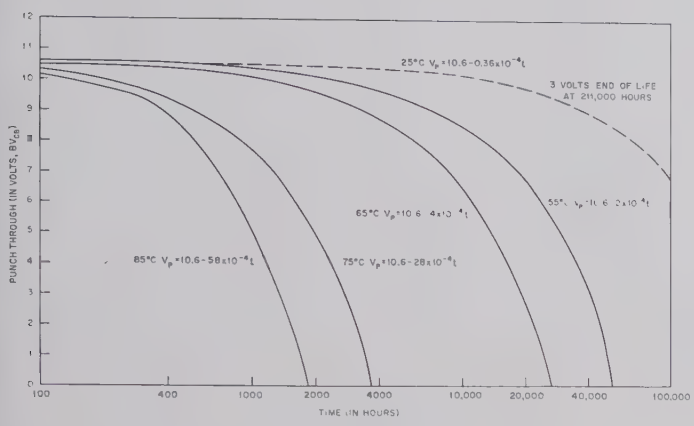


Fig. 4—Punch-through voltage accelerated life test.

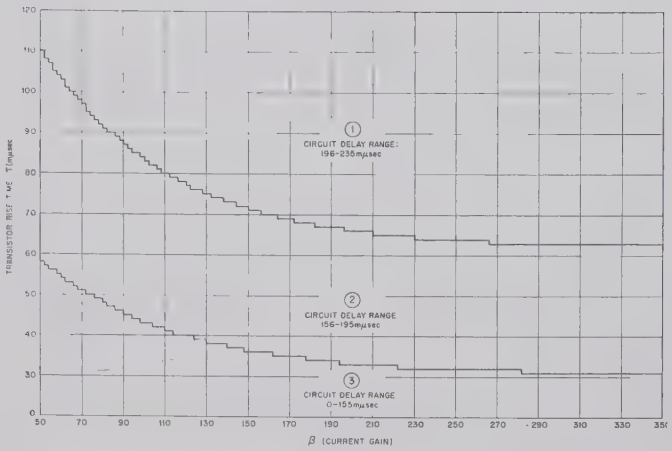


Fig. 5—Circuit delay ranges.

Close cooperation with the manufacturers was maintained to insure that 1) the transistors received were the best that could be produced in the manufacturing process used, and 2) to insure uniformity among transistors produced at different times. Statistical studies were also conducted with the UNIVAC I computing system in order to correlate the transistor parameters to the changes in production control (such as germanium resistivity, etching time, and so forth).

Another study [12] conducted with the aid of the UNIVAC I system was the correlation between transistor parameters and the delay exhibited by the transistor when it is inserted in the circuit. This correlation was necessary because functional tests of transistors operating in appropriate circuits do not directly aid the manufacturer in determining what combination of parameters caused the transistors to fail. One such study involved the test on 200 transistors selected from different groups. The study established functions correlating the delay to the circuit parameters and resulted in:

$$\text{delay} = K_1T + K_2T^2 + K_3VT + K_4T^2/\beta + K_5S/\beta + K_6$$

where

T = rise time
S = storage time
V = base to emitter voltage with transistor on
β = beta (dc large signal current gain).

A program was written to apply a least-square fit to the 200 sets of transistor data for the given equation. Thus, the values of the constants K_1–K_6 were determined. Fig. 5 shows the dependence of the delay on rise time (T) and beta (β) only.

Evaluation of Circuitry

The general types of circuitry that can be classified are the following.

Pulse Envelope (Nonreturn to Zero)

Four basic types can be selected:

1) One as opposed to two levels of logic in the basic logical block. (The existence of more than two levels becomes impractical due to gain and speed required.)
2) Inversion to be included, or not, in basic logical block (that is, grounded emitter as opposed to two grounded emitters or an emitter follower).
3) Diode logic as opposed to transistor logic (that is, whether diodes are to be used or not) [6].
4) Saturated or nonsaturated transistor operation (that is, whether or not delays are to include storage time).

The four above choices indicate 16 types of circuitry. In addition, each type can be represented by a number of possible circuits.

Return to Zero

Since square-loop magnetic devices were rejected because of low gain and speed, only transformer transistor circuitry can be considered. Here, two major types of circuitry exist:

1) Feedback circuitry (blocking oscillators, and so forth) fit for timing, storage, and pulse forming with diode logic.
2) Transformer logic [8], [9].

In order to evaluate all types of circuitry to be used, a test system which could simulate the actual system was built. For the results to be critical, each test system should be designed with as much care as the final system. Since this would involve a prohibitive amount of work, six test systems, which simulate the work of complete adders, were built. The design of these adders may not be optimum, although the validity of the results depends on each design being truly representative of the capability of the type of circuitry. The result of each design consisted of:

1) Addition time. This may differ considerably from the desired $\frac{1}{2}$-μsec period, since the design may

optimize speed irrespective of the specification requirement.
2) Cost of components.
3) Number of components having relatively low reliability (such as transistors, diodes, and other special devices).

The number of components, cost, and addition time can be combined into a sort of merit factor in order to represent uniquely the advantage of one circuit as opposed to another. Of course, such a merit factor may assign erroneous weights to the various performance numbers, and therefore overreliance on a merit factor should be avoided.

For example:

Factor	Symbol
Cost of parts	C in \$
Addition time	T in μsec
Number of transistors	N_t
Number of diodes	N_d
Merit factor	M

Since a diode was considered three times more reliable than a transistor,

$$M = \left(\frac{1}{TC}\right)\left(\frac{1}{N_t + N_{d/3}}\right).$$

In addition there are a number of other factors which are designated intangibles. The following is a list of some of these intangibles:

1) Possible forms of packaging.
2) Labor in packaging.
3) Power and clock tolerances and cost.
4) Frame construction.
5) Noise pickup.
6) Debugging and service.
7) Marginal checking.

Another important factor to be considered is the type of circuitry with which the project personnel are most familiar. By comparing the performance of a standard circuit with the circuit being developed, the circuit offering the greatest advantage was chosen. The process described above was followed and six complete 4-digit adders were constructed. Three adders were constructed with circuitry similar to those described in [6], [8], and [9], respectively. One adder used the circuitry of Fig. 6, another used circuitry similar to Fig. 6, only with two levels of diode gates, and the other adder used a variation of the circuitry described in [7]. In sum and without going further into a description of the test system process, the pulse-envelope type of circuitry was selected employing single-level diode logic and saturated transistor operation with inversion. This decision was based on the evaluation of the merit factor for each one of the adders and after consideration of the intangibles as described above.

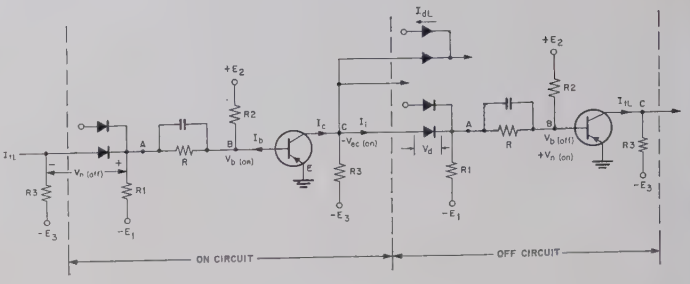


Fig. 6—Two stages of the basic gate circuit.

The Worst Case Design and Optimization of a Circuit

After all the types of circuitry have been evaluated and one type is chosen, the process can become more rigorous and thorough.

The worst case for dc will be discussed first. Fig. 6 shows the circuit for two gate inverters. The transistor in the first circuit is turned on and produces a large collector current. The transistor in the second circuit is turned off. The ON and OFF conditions of the transistors can be defined as follows.

ON condition:

Emitter-collector voltage $(V_{ec}) = V_{ec(on)}$
Output current $= n \times I_i$ (n = fan-out of three)
Input voltage $(V_{i(off)} = E_3 - I_{tL}R_3 (I_{tL} \leq 250$ μa at 30°C)
Emitter-base voltage $(V_{eb}) = V_{b(on)} \leq 0.6$ volt.

OFF condition:

Emitter-base voltage $(V_{eb}) = 150$ mv, $I_b = 0$, and $I_c = I_{tL} (I_{tL} \leq 250$ μa).

Since V_b vs I_b is nonlinear (Fig. 7) and varies greatly from transistor to transistor, it is difficult to make a general statement on the precise values of V_b and I_b to be used in design. The best over-all circuit performance was obtained (using a surface-barrier transistor). When the design was based around the off point ($I_b = 0$), this off point was optimum because this transistor has a relatively low beta value for very small base current. The $I_b = 0$ point corresponds (at 25°C) to a collector current of less than 250 μa.

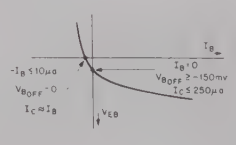


Fig. 7—V_{eb} vs I_b for typical transistor.

The worse case equations for nodes A, B, and C are shown in Table I.

At node A of the ON circuit, the diode should be reverse biased to assume a zero-input current. [That is, the voltage at node C of the OFF circuit plus the noise

TABLE I
TABLE OF EQUATIONS

Node		Equation
OFF	ON	
C	A	$E_{3(\min)} - I_{tL(\max)}R_{3(\max)} - V_{n(\text{off})} = \dfrac{E_{1(\max)} - V_{b(\text{on max})}}{R_{1(\min)} + R_{(\max)}} R_{(\max)} + V_{b(\text{on max})}$
	B	$I_{b(\text{on min})} = \dfrac{E_{1(\min)} - V_{b(\text{on max})}}{R_{1(\max)} + R_{(\max)}} - \dfrac{E_{2(\min)} + V_{b(\text{on max})}}{R_{2(\min)}}$
	Transistor	$I_{b(\text{on min})}\beta_{(\min)} = I_{c(\max)}$
	C	$I_c = \dfrac{E_{3(\max)} - (V_{ec(\text{on max})} + V_{d(\max)})}{R_{1(\min)}} + nI_{i(\max)} + mI_{dL(\max)}$ n = fan-out; m = number of diodes in driven gates
A		$I_{i(\max)} = \dfrac{E_{1(\max)} - (V_{ec(\text{on max})} + V_{d(\min)})}{R_{1(\min)}} - \dfrac{(V_{b(\text{off})} + V_{n(\text{on})} + V_{ec(\text{on max})} + V_{d(\min)})}{R_{(\max)}}$
B		$V_{b(\text{off})} = \dfrac{1}{R_{2(\max)} + R_{(\min)}} [E_{2(\min)}R_{(\min)} - I_{b(\text{off})}R_{(\min)}R_{2(\max)} - R_2(V_{n(\text{on})} + V_{ec(\text{on})} + V_{d(\max)})]$

voltage induced in the input line ($V_{n(\text{off})}$) should not be positive with respect to the voltage at node A of the ON circuit.]

At node B of ON circuit $I_{b(\text{on, min})}$ and $I_{c(\max)}$ should be maintained.

At node C of the ON circuit, $I_{c(\max)}$ should exceed load current (nI_i), diode-leakage current (mI_{dL}), and the current into R3.

The voltage at node B of the OFF circuit should ensure that the transistor is turned off ($I_{b(\text{off})} \leq 0$). In Table I, there are six equations with 20 unknowns. These unknowns (dropping maximum and minimum subcripts) are:

I_{tL}, I_{dL}, $I_{b(\text{on})}$, $I_{b(\text{off})}$, I_c, I_i, and β
E_1, E_2, E_3, R_1, R_2, R_3, and R
$V_{n(\text{on})}$ and $V_{n(\text{off})}$
$V_{b(\text{on})}$, $V_{b(\text{off})}$, $V_{ec(\text{on})}$, and V_d.

Obviously, for defining a unique circuit, it is necessary to determine 14 independent unknowns in the process of optimization.

$V_{b(\text{on})}$, $V_{b(\text{off})}$, V_d, I_{tL}, I_{dL} and $I_{b(\text{off})}$ are determined so that a high percentage of the produced diodes and transistors can be used. $V_{n(\text{on})}$ and $V_{n(\text{off})}$ are determined through noise measurements (which will be discussed further).

As a zero approximation for the solution, we assume $I_{b(\text{on})}$ to be such as to allow operation in the range where beta is maximum (upper end of this range). This assumption should be re-examined at the end of the optimization because iterative corrections of the value of $I_{b(\text{on})}$ may be necessary. $V_{c(\text{on})}$ can be determined by assuming values for four independent variables and calculating (from equations in Table I) the relationship of β vs $V_{c(\text{on})}$. On the other hand, transistor measurements give the available β vs $V_{c(\text{on})}$ for the end of transistor life (aged transistor). Comparison of the rates of change of beta with $V_{c(\text{on})}$ (expressed as $d\beta/dV_{c(\text{on})}$) of the two curves shown in Fig. 8, allowed for the determination of an optimum value of 0.3 volt for $V_{c(\text{on})}$.

E_2, so long as it exceeded 10 volts, will have little effect on the operation of the circuit. To minimize power supply requirements, 12 volts was selected for the value of E_2. Consequently, six equations with nine unknowns still need to be solved. The remaining independent unknowns are determined through optimization of speed.

Maximum Delay Situation

The delay of a step function through a transistor depends on whether it is turning off or on, since the worst loading conditions are different for the two switching directions. Therefore, delay is conveniently measured across two levels; one switching in one direction and the other switching in opposite direction (each being loaded for its own worst case of delay). When the transistor is turning on, its beta and beta cutoff frequency should be the minimum allowable, its load should be maximum, and the worst diode leakage should be present.

When the transistor is turning off, the worst condition occurs when the load is minimum and the beta and storage time are maximum. This condition oversaturates

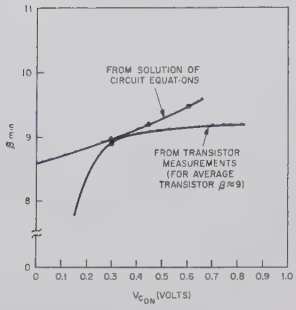


Fig. 8—Beta vs $V_{c(on)}$.

the transistor as much as possible, thereby producing the maximum hole storage delay. If hole storage did not constitute an appreciable portion of the total delay for the SB-100 transistor, the worst loading condition for the transistor turning off would have to be re-examined.

Measuring Technique

The breadboard setup for delay measurements is shown in Fig. 9(a). The first gate gives input rise and fall times comparable to what would be expected in a chain of logic. To simulate wiring capacity, 50 $\mu\mu$f of lumped capacity was added from all collectors to ground. Diode and transistor leakage were also added where appropriate.

The delay was measured from collector to collector, usually 1 volt below ground, since this is approximately where the following stage is fully turned on during the transient period. If the trailing edges of the waveshape at the first and third collectors are identical, it does not matter from which point on the trailing edges the delay is measured. However, if the trailing edges are not identical the delay measured between a fixed point on the two edges will be either pessimistic or optimistic, depending on whether the output fall time to the point of measurement is greater or less than the corresponding fall time on the driving edge.

Standard delays were measured by observing the recirculation frequency of a chain of amplifiers shown in Fig. 9(b). The odd level of gating must be added in order to make the chain unstable. Here the minimum load is one-third. For a given delay per stage, any desired pulse width may be obtained by varying the number of stages. This method of measuring delay has the advantage that the input and output waveforms of any given stage are assured of being those which would be expected in a long chain of logic. In this type of experiment, wiring capacitance must be simulated.

As shown in Fig. 10, E_3 can be determined by designing circuits for a number of E_3 values and then measuring the delays. The upper value of E_3 is limited by the breakdown voltage of the transistors. For longer transistor life, E_3 was selected to be 3 volts. The graph shows that 4 volts for E_3 would only improve the speed slightly.

The next parameter to be optimized was the value of R_3. This is shown in Fig. 11, where the delay is plotted

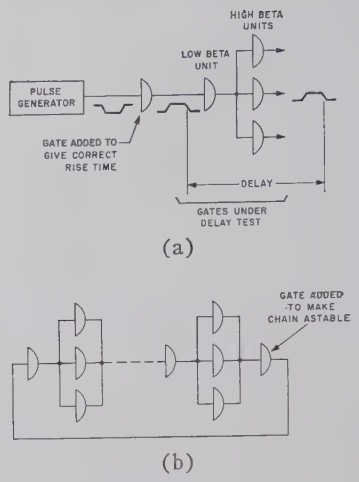


Fig. 9—Block diagram for delay measurements.

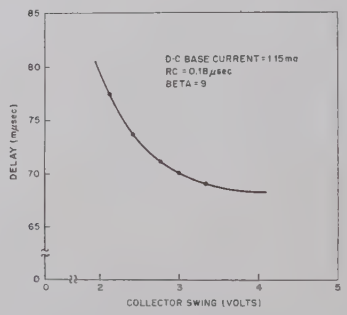


Fig. 10—Delay as a function of collector swing.

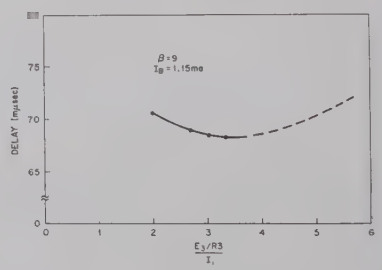


Fig. 11—Delay as a function of the current through R3.

as opposed to the current in R_3. The optimum speed is obtained by

$$\frac{E_3/R_3}{I_i} = 3.3.$$

The only remaining independent variable is beta. Although speed can be increased with the increase in beta, the value selected for beta was 9 as the end of transistor life. As directed by the specifications, this is the minimum allowable beta for assuring a maximum delay of 40 mμsec per level of logic with three loads. The value of the capacitor should now be varied to obtain maximum speed for all pulse widths propagated through the levels of gating.

The process of optimization requires the design of a large number of circuits (less than 100). Each design implies solution of six simultaneous nonlinear equations.

The amount of computations involved clearly indicates the advantage of processing the computations on a computer.

Transient analysis of circuit performance was conducted experimentally (as shown by the delay measurements). We have developed mathematical techniques for transient analysis [13]. However, the results of such analysis were qualitative and helpful in understanding the factors involved, but were not sufficiently accurate for the design of the circuits.

Statistical techniques and a library of computer routines have been developed for correlating circuit delay with transistor parameters, which facilitated the preparation of transistor specifications (see Fig. 5).

Noise

The subject of signal and dc distribution noise will be discussed in this section.

Signal noise is studied to determine the values of $V_{n(on)}$ and $V_{n(off)}$, which are the margins allowed to prevent the amplification of noise picked up in the signal leads that connect the gates (Table I). The dc distribution noise consists of part of the tolerances of the dc supply voltages which should be determined. Computation of the generated noise is practical, since this problem has been treated extensively in telephone line applications [14]. Since the computer operates at frequencies about 10^5 times higher than telephone lines, the pickup amplitude per unit length is correspondingly greater. The margins against signal noise can also be obtained by experiments, some of which will be described in this paper. (They also are the model for similar results through computations.)

Signal noise is a result of induction from other wires in the vicinity of the wire in which information pulses are propagated. Signal noise can also be a result of ringing caused by variations in load impedances and unmatched terminations. The induced currents and voltages are caused by either mutual inductance or capacitance. We found that the capacitive and inductive pickup are opposite in amplitude, with the latter being the larger. Therefore, we are mostly interested in inductive pickup.

The response of the gate to noise depends on both the amplitude and frequency of the noise. The test setup illustrated in Fig. 12 is designed to establish the amplitude and frequency for the worst case of noise propagated through a gate. First, we established that the worst case corresponds to the condition where the driver circuit is on and heavily saturated. This condition represents almost a short circuit at the driving end of the pickup line. The driven circuit represents the smallest load possible, and consists of a high-speed transistor which is lightly loaded at its output by a chain of gates. Voltages and components were also picked for the worst case. The lead between the driver and the driven circuit is bundled together with leads originating from a number of other gates which are in the process of being turned on and off

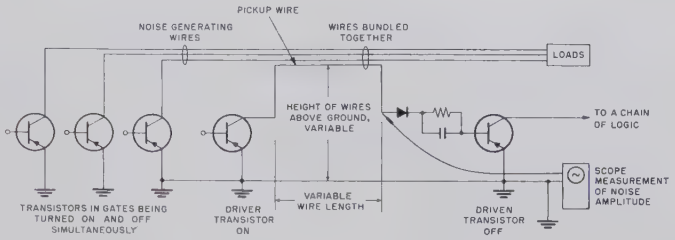


Fig. 12—Experiment for noise measurements.

(these circuits have rise time of less than 20 mμsec). The other ends of the wires, on which pulses are being propagated, are connected to various loads.

This experiment led to the following conclusions [15]:

1) Noise generated in the pickup wire is directly proportional to the legnth of the line and to the logarithm of the height of the wires above ground.
2) Noise will decrease with an increase in the distance between the pickup wire and the generating wires.

In order to minimize noise, we used 30-gauge wire and made the TEFLON[1] insulation thick to assure maximum spacing between wires. We found that in the case of the seven tightly bundled wires from the basic gates, the noise generated in the pickup wire (up to 15 inches long) had an amplitude below 400 mv. The total noise generated was divided between the backboard wiring and the wiring on the circuit packages. We allowed a 400-mv pickup noise on the backboard and 100 mv on the cards, giving a total of 500 mv of pickup noise [this corresponded to $V_{n(on)}$ and $V_{n(off)}$ of 150 mv]. We also found that noise is reduced appreciably when it is propagated through a chain of gates.

Upon studying the distribution of the intra-unit wire length, we found that a great majority of wires were less than nine inches long. Therefore, it was feasible, without undue expense, to allow point-to-point signal-lead wiring for all wires up to 9 inches long (leaving a 6-inch margin for safety). For longer wires, we used twisted pairs in which one wire was the signal lead and the other was connected at its ends to the ground points at the driving and driven circuits. We found that the mutual inductance per foot of single leads can be as high as 1 μh per foot. However, the mutual inductance of a twisted pair is reduced to 10 mμh per foot. Inter-unit leads used coaxial cables where the mutual inductance is of the order of 1 mμh per foot. These studies showed that noise could be suppressed (at a sacrifice in cost) by using more expensive wires, such as twisted pairs and coaxial cables.

Since noise amplitude depends on the distance of the wires from ground, all metal parts in the circuit card frames and the main frame consitute the ground system. In order to assure good ground conduction at all points (without worry of corrosion), gold plating was used in

[1] TEFLON is the trademark of E. I. du Pont de Nemours & Co., Inc.

many places. This concept of a common ground is radically different from the concept of grounding braids. By employing this method of grounding, the margins for noise for which allowance is made in the design were much smaller than the noise margins in previous computers. This allowed us to design considerably more efficient circuits in terms of larger gain and shorter delays.

Ringing on signal leads is caused by unmatched terminations. It is difficult to terminate the sending end of the wire, since the transistor impedance changes greatly when it is turning on. The impedance of a single wire is approximately 300 ohms; the impedance of a twisted pair is approximately 200 ohms. The minimum terminating impedance at the receiving end of the leads was approximately 500 ohms and the maximum was approximately 1500 ohms. Smaller gate input impedances were desirable, but this was not possible because of the relatively low current level of the transistors being used (about 10 ma). The ringing noise resulting from the unmatched termination was still smaller than the signal noise; therefore, it did not constitute a problem in this case. Generally, in the case where a choice is possible, lower gate input impedances resulted in reductions in ringing, noise amplitude, and circuit delay.

Another source of noise occurs when a number of circuits share a common ground impedance. However, the latter proved to be negligible because of the precautions taken in designing the ground system.

The system of backboard wiring required an examination of its effect on the delays in propagation of information. We found that for intra-unit wiring, distances were always smaller than six feet. Since rise times were between 10 to 60 mμsec, computation based on lumped circuitry, rather than distributed circuitry, provided reasonable accuracy. Therefore, in our delay measurements, we simulated the length of wires by attaching a capacitance of 6 μμf per foot to the collectors of the transistors.

Noise on DC Distribution Lines

The transmission lines carrying dc voltages to the circuit should be checked for noise that may be generated in them. A typical transmission line is made up of the equivalent lumped components shown in Fig. 13 [16]. A low-impedance battery is connected to a bank of capacitors (C1), which is connected through a lead (inductance L3) to a very low-impedance transmission line. Because of the short distances in the computer, intra-unit transmission lines can be simulated by a lumped inductance (L4) and capacitance (C2). The wires leading to the cards are capable of picking up noise (G_n) and having equivalent inductance (L5). It is possible to place capacitors (C3) on the circuit cards which also have a certain amount of lead inductance (L7) and resistance (R3). In addition, the printed wiring inductance (L6) must be considered.

There are two sources of pickup noise to be considered.

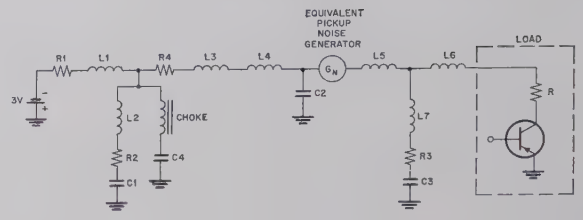


Fig. 13—Equivalent dc distribution circuit.

One is attributed to the turning on and off of transistors, which can account for currents as great as 50 amperes, which have a rise time of about 150 mμsec. With current steps of such magnitude, the equivalent circuit in Fig. 13 has to be examined for the impedance encountered over the entire line. The amplitudes of the voltages induced can be calculated and checked experimentally. The other source of pickup noise (G_n) is encountered by the leads to the circuit cards. Means should be provided for dissipation of transient energy in the dc distribution system. Since the resistance of the line may not be sufficient to damp the ringing, additional resistors between the distribution system and ground must be added. However, this method is costly in terms of the wasted power. Instead, the ac power could be dissipated in capacitors or magnetic chokes having large hysteresis loops. Such a dissipation is accomplished by the choke and capacitor (C1) shown in Fig. 13. The approximate values of the lumped components are given in Table II.

TABLE II
EQUIVALENT CIRCUIT VALUES

Lumped Component	Description	Order of Magnitude
L1	Power supply lines	10 μh
L2	Capacitance (C1) lead	1 μh
L3	Leads to transmission lines	1 μh
L4	Transmission line	1 mμh
L5	Lead to circuit card	100 mμh
L6	Printed wiring on card	10 mμh
L7	Capacitor lead (C3)	10 mμh
C1	High-loss capacity	farads
C2	Transmission line	1000 μμf
C3	Capacitor on circuit card	10 μf
C4	Choke capacitor	100 μf
R1, R2, R3, R4	Lead resistances	0.1–1 ohm
R	Equivalent resistance of total load	60 milliohm

Figs. 14 and 15 illustrate the wiring and power distribution schemes. Fig. 14 is a photograph of the backboard of the LARC computing unit. The heavier vertical lines are the dc and clock distribution buses. The very dense point-to-point wiring (entirely covering the printed circuit connectors) can be seen between the buses. Fig. 15 is an enlargement of the top of the backboard. On the left are shown the heavy power supply cables; the two rows of cylinders are the large capacitor banks. These are connected to the vertical dc buses projecting above the backboard. The leads from the buses to the circuit cards are shown at the lower right corner of Fig. 15.

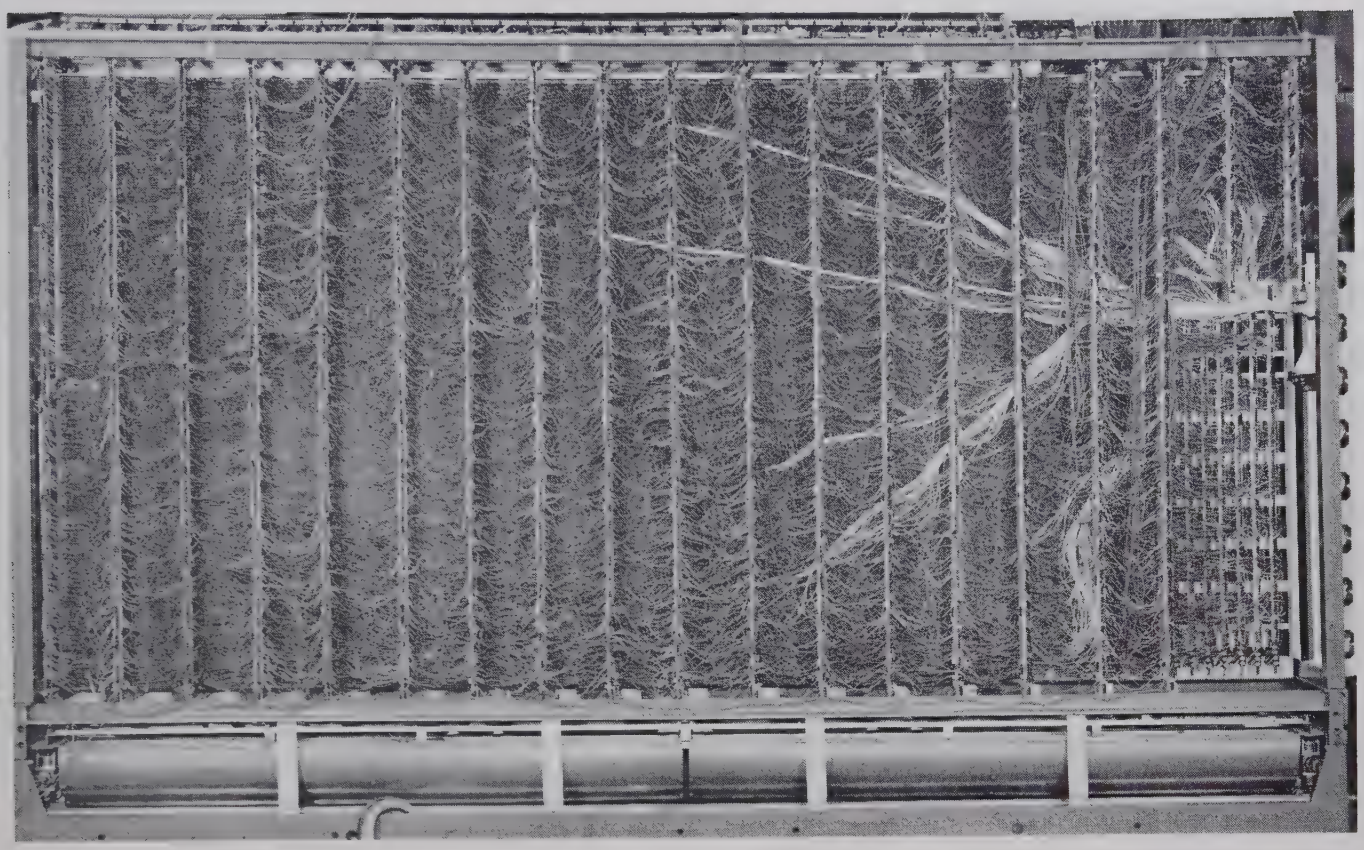


Fig. 14—Backboard of LARC computing unit.

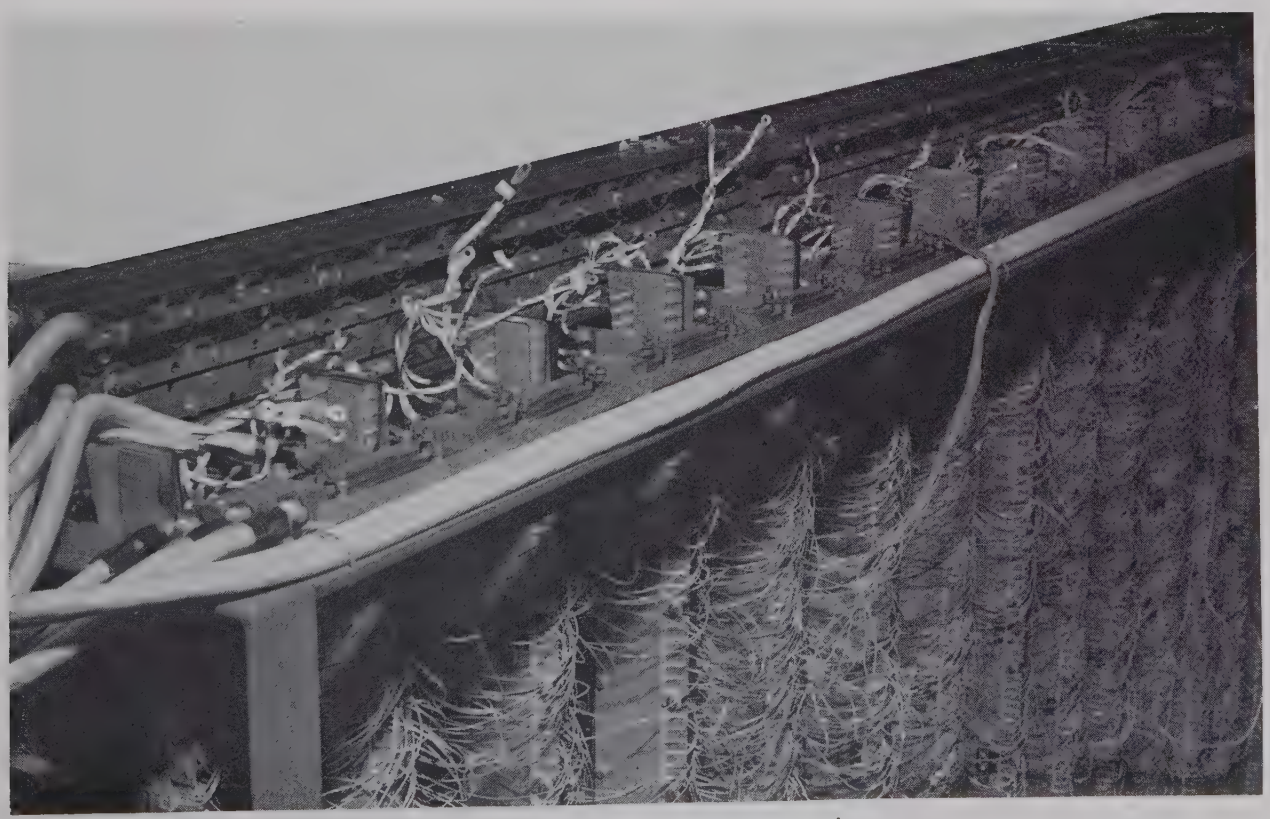


Fig. 15—LARC computing unit, top rear view.

Pulseformers and Timing

The following is a description of how the gate circuit fits the general intra-unit scheme.

In the logical design of the LARC system, timing is an extremely important consideration since all operations must be synchronized. In the process of generating the signals needed to perform the various logical functions, an information level is conveyed through a path or several paths. Each path consists of many stages of transmission units (gates, delay elements, and so forth). No unit can be operated without introducing some delay, which is not constant but varies within a range depending on the tolerances allowed for the passive and active elements used in the unit. Consequently, a means must be provided for synchronizing the various signals with the central clock pulses of the system so that the output of one stage or logic level reaches some other stage at the correct instant in time. The pulseformer is the standard LARC circuit used to synchronize the changes in information level being propagated throughout the system.

The basic pulseformer circuit is composed of three parts: a gate-inverter driver, a bidirectional transmission gate and a flip-flop [see Fig. 16(a)]. The gate inverter performs a logical function at the input of the pulseformer and acts as the driving element for the bidirectional gate and flip-flop circuits. The bidirectional gate consists of a pair of diode AND gates which are capable of handling both polarities of clock pulses as control inputs, and in furnishing a positive or negative pulse as triggering input to the flip-flop. The flip-flop circuit is composed of two transistor amplifiers connected to form a feedback loop.

The complete schematic diagram of the pulseformer circuit is given in Fig. 16(b). The principle of operation of the driver circuit is similar to the basic gate circuit described previously. The circuit parameters have been chosen so that the input requirements are the same as the basic gate while the output supplies the correct level and polarity of drive for controlling the bidirectional gate.

Basically the bidirectional gate consists of a positive AND gate (diodes CR4 and CR8) and a negative AND gate (diodes CR6 and CR9) whose function is to sample the output of the driver stage during the 100-mμsec period of the clock pulses. At all other times, the gates are logically closed so that no change will take place at the output at S, that is, at the junction of the isolating diodes CR5 and CR7. Therefore, if the output of the driver is positive at the time of a clock pulse, the positive clock

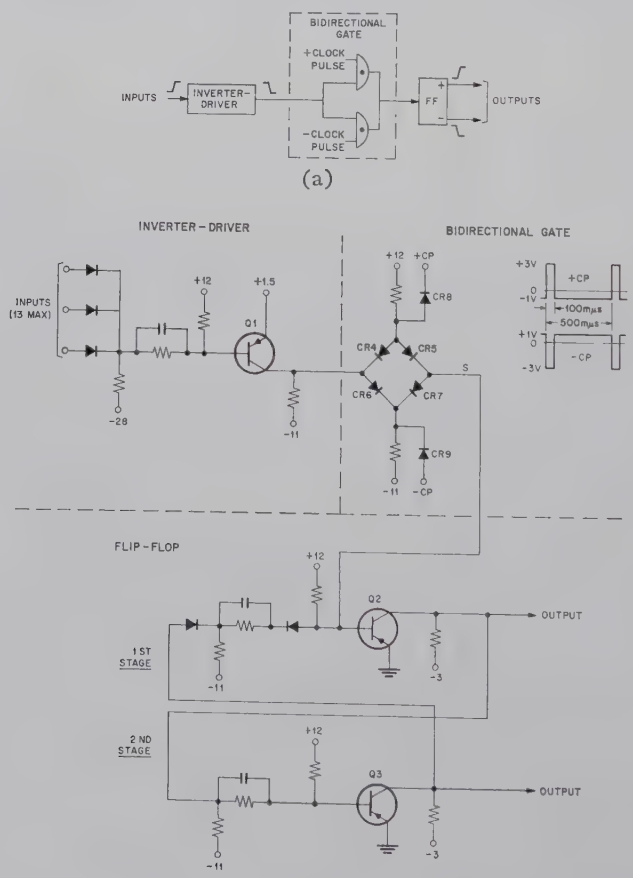


Fig. 16—Pulseformer circuit.

gate (CR4, CR8) must function so as to give a positive-going output at S and trigger the first stage of the flip-flop from the on state to the off state, or if Q2 is already off, to keep it off. Similarly, if the driver output is negative at the clock-pulse time, the negative clock gate (CR6, CR9) must operate so that the output is negative-going and reverses the state of the flip-flop.

Since the purpose of the pulseformer is to synchronize signals at various points in the logical network, the output of a pulseformer must transmit through no fewer than a certain minimum number of stages and no more than a certain maximum number of stages before it is sampled (retimed by the synchronizing clock pulses) by the next pulseformer. The number of stages allowed between successive pulseformers is determined by the sampling rate (clock-pulse frequency: 2 Mc), clock-pulse width (100 mμsec maximum), and the maximum and minimum delays of the pulseformer and the intervening logical network. With the standard unit of delay defined as that of the basic gate (minimum unit 11 mμsec, maximum unit 40 mμsec), the minimum and maximum number of stages or levels of circuit logic—in terms of delay units—that can be used between pulseformers must be 6.6 to 9 units. This is depicted in the general block diagram of Fig. 17. The maximum delay time between pulseformers should be less than the period of 2 Mc (500 mμsec). The waveform timing diagram in Fig. 18 indicates that this condition is met when due allowances are made for all circuit delays. The maximum time by which the input to the pulseformer driver gate must precede the next clock pulse is a function of the width of the clock pulse. The required precedence increases with a decrease in the clock pulse width. The minimum delay condition is illustrated in Fig. 19. A change in the output of a pulseformer which feeds a chain of logic should be reflected 500 mμsec later at the output of the pulseformer at the other end of a chain of logic. However, if the delay between pulseformers is too short, the information will travel fast enough so that both pulseformers are triggered by the same clock pulse resulting in erroneous operation. This is prevented if the minimum delay between pulseformers plus the maximum allowed overlap is longer than the pulse width plus any jitter.

Both minimum and maximum delay conditions are met for a clock pulse of minimum width of 80 mμsec and maximum width of 100 mμsec. If the width of the clock pulse is reduced any further, more margin could be allowed at the minimum delay conditions. For each machine, the nominal pulse width could be determined experimentally for maximum margins at both ends. This means that if at some future date the characteristics of the transistors improve to allow shorter delays, a narrowing of the clock pulse and increase of the frequency would be possible.

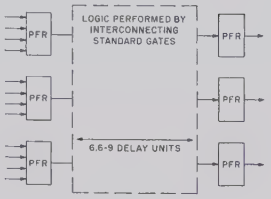


Fig. 17—Block diagram of logical-network chain.

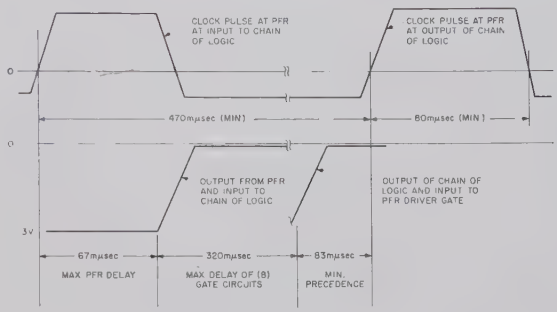


Fig. 18—Maximum delay condition.

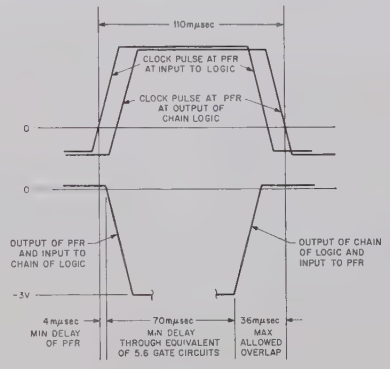


Fig. 19—Minimum delay condition.

Conclusion

At the outset of the design of the LARC computing system, we made reliability estimates of the computer. Among other components, the LARC computing unit has 90,000 diodes, 27,000 transistors, and 30,000 capacitors. From the available (1956) component distribution and life data [17], we estimated an average of 16 hours of failure-free operation (the data in Table III represent accumulative Univac experience up to 1956). Operating experience indicates that operation time without failure will be in the vicinity of 200 hours. This improvement was achieved through careful component and circuit design.

In addition, the check-out time for the computing unit has been surprisingly short (three months) for its complexity. The relatively short period of testing was largely due to the simulation of logic by the UNIVAC I system, and preparation and continuous updating of the wiring lists. However, the reliability of the circuits was also very gratifying.

TABLE III

COMPONENT RELIABILITY STUDY*

Component	Quantity	Remington Rand Univac Study	
		Failure (per cent)	Number of Failures
Tubes	20	4.0	0.8
Transformers	10	0.8	0.08
Diodes	90,000	0.008	7.2
Printed Circuit Cards	4000	0.13	5.2
Transistors	27,000	0.036	9.7
Ceramic Capacitors	30,000	0.01	3.0
Tubular Capacitors	5000	0.01	0.5
Resistors	120,000	0.0075	9.0
Connectors	4200	0.25	1.05
Relays	50	4.0	2.0
Motors	5	20.0	1.0
Ferractors	6000	0.1	6.0
Power Supplies	10	33.3	3.3
Leads with Connections	180,000	0.002	3.6

* Data represent accumulative Univac experience up to 1956.

The work described in this paper constitutes only a small portion of the total circuit design of the LARC computing system. To give a detailed description of the other circuits optimized would be beyond the scope of this paper.

ACKNOWLEDGMENT

The authors gratefully acknowledge the invaluable assistance and suggestions given by the members of the Component and Standards and the Applied Mathematics Departments of Univac Engineering Center, Philadelphia, Pa.

REFERENCES

[1] J. P. Eckert, J. C. Chu, A. B. Tonik, and W. F. Schmitt, "Design of Univac®-LARC System: I," *Proc. Eastern Joint Computer Conf.*, Boston, Mass., December 1–3, 1959, pp. 59–65.
[2] H. Lukoff, L. M. Spandorfer, and F. F. Lee, "Design of Univac®-LARC System: I," *Proc. Eastern Joint Computer Conf.*, Boston, Mass., December 1–3, 1959, pp. 66–74.
[3] J. P. Eckert, "Univac-LARC, the next step in computer design," *Proc. Eastern Joint Computer Conf.*, New York, N. Y., December 10–12, 1956, pp. 16–19. (AIEE Special Publication T-106.)
[4] W. F. Schmitt and A. B. Tonik, "Sympathetically programmed computers," *Proc. Internatl. Conf. on Information Processing*, Paris, France, June, 1959, pp. 344–347.
[5] Private communication from M. A. Breslin (Remington Rand Univac, Philadelphia, Pa.).
[6] R. H. Beter, W. E. Bradley, and R. B. Brown, "Surface-barrier transistor switching circuit," 1955 IRE CONVENTION RECORD, pt. 4, pp. 139–145.
[7] H. S. Yourke, "Millimicrosecond transistor current switch-circuit," IRE TRANS. ON CIRCUIT THEORY, vol. CT-4, pp. 236–240; September, 1957.
[8] T. H. Bonn, "Analysis of magnetic amplifier circuits," *Proc. Internatl. Symp. in the Theory of Switching, April, 1957: Ann. Computation Lab. Harvard Univ.*, vol. 30, pp. 149–160; 1959.
[9] N. S. Prywes, "Coil Gating," Remington Rand Univac, Philadelphia, Pa., Tech. Monograph; August, 1957.
[10] Private communications from G. Kaskey (Remington Rand Univac, Philadelphia, Pa.).
[11] "Statistical Techniques in Transistor Evaluation," Remington Rand Univac, Philadelphia, Pa., Final Rept., prepared under Contract No. NObs-72382; April, 1959.
[12] Unpublished paper by G. Kaskey and N. S. Prywes (Remington Rand Univac, Philadelphia, Pa.).
[13] Private communication from H. J. Gray (The Moore School of Elec. Engrg., University of Pennsylvania, Philadelphia).
[14] A. G. Chapman, "Open wire crosstalk," *Bell Sys. Tech. J.*, vol. 13, pp. 19–58, 195–239; April, 1934.
15] Private communication from D. Benima (Remington Rand Univac, Philadelphia, Pa.).
[16] Private communication from R. Torrey (Remington Rand Univac, Philadelphia, Pa.).
[17] Private communication from H. J. Gray (The Moore School of Elec. Engrg., University of Pennsylvania, Philadelphia).
[18] Private communication from W. F. Schmitt (Remington Rand Univac, Philadelphia, Pa.).

Coincident-Current Superconductive Memory*

L. L. BURNS, JR.†, SENIOR MEMBER, IRE, G. A. ALPHONSE†, MEMBER, IRE, AND G. W. LECK†

Summary—**In a continuous superconductive film memory, elements are obtained through discrete regions of circulation of persistent currents near the intersection of *x-y* conductors deposited on the film. Analysis and confirming experiments show that these regions are stable. The elimination of edges of discrete film dots removes the main cause of variation of critical currents. Reproducibilities better than one quarter per cent were obtained. Simplicity of construction permits high bit densities. Memory planes of one hundred cells were made. Advisable speed of operation depends mainly on addressing and sensing circuits. Write-in in 3 nsec was obtained in single elements with only 60 milliamperes drive.**

* Received by the PGEC, March 6, 1961.
† RCA Corp., Princeton, N. J.

INTRODUCTION

FOR the past several years various research laboratories have carried on work on the practical application of superconductivity and in particular on superconductive computer components. This effort has shown clearly that a serious limitation of present-day computers can be overcome by proper application of the phenomenon of superconductivity. Practically all present-day computers of general utility are memory-limited; *i.e.*, the complexity of the mathematical operations that these computers can perform is restricted because of the size and speed of the memory.

The results obtained thus far on superconductive memories by the various research organizations have shown that memory operation is possible, that the phenomena are adequate for the desired purpose, and that small volume per memory bit stored can be achieved. However, fabrication difficulties and a lack of understanding of the parameters involved in reproducibility have prevented the development of really large-scale memory arrays. This paper describes an approach that overcomes most of the difficulties in fabrication and reproducibility through the use of a continuous superconductive film instead of discretely fabricated elements.

Superconductivity

Superconductivity is that property of certain metals of exhibiting a complete loss of electrical resistance at low temperatures. The loss of resistance is complete; *i.e.*, the resistance is absolutely zero. This implies certain very interesting and useful properties for superconductors. For example, if a superconductor forms a part of an electric circuit, any alteration in magnetic field would excite suitable persistent surface currents so as to maintain the internal magnetization constant. (An additional fact which cannot be derived from the criterion of zero electrical resistance is that, while it is true that the magnetic induction within the superconductor is time invariant, it has been found that it is actually exactly zero at all times.) Thus, ideally the superconductive state is not only infinitely conducting but also perfectly diamagnetic.

An additional phenomenon of superconductivity is that the superconductive state is destroyed when a sufficiently strong magnetic field is imposed on a superconductor, and as a consequence of this, normal resistance returns abruptly to a superconductor which is carrying a sufficiently large current. A very weak magnetic field will restore the resistance of a superconductor when the temperature is only slightly below the critical temperature, whereas a stronger field is required at a lower temperature. In general, the relation of the required field strength H_c and temperature is approximately parabolic, as shown in Fig. 1, for a number of known superconductors.

Materials

There are a great many materials that display the phenomenon of superconductivity. Table I lists the elements that are known to be superconductors and Table II lists a number of well-known superconducting compounds. Despite the relatively large number of superconducting materials the choice for use in computer memories and logical circuits is quite limited. Hard materials have characteristics that are not desirable in computer circuitry. The fabrication techniques that appear to be the most promising, evaporation and plating, limit the materials that can be used to those that are suitable for these techniques. Also, the temperatures that are realistic for simple closed-cycle refrigerators limit the choice to those materials that are superconducting near 4.2°K. The materials of choice at the present state of the art are tin and lead. Both are easy to handle, easy to evaporate, and easy to plate. The efficiency of tin as a gate material for cryotrons approaches 80 per cent (ratio of actual critical current to theoretical critical current). Lead has a high enough critical temperature to be useful as a control in cryo-

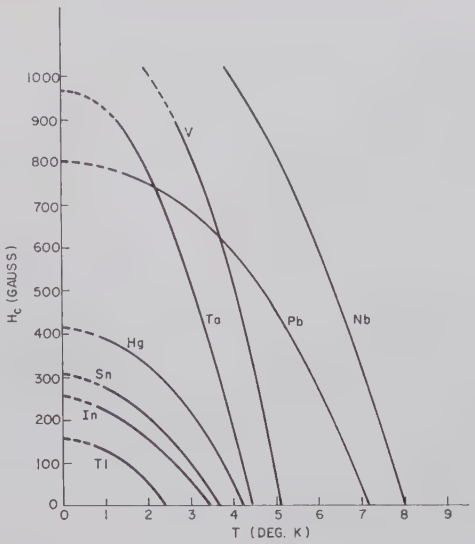


Fig. 1—Critical-field vs temperature curves (H_c-T) for various metals.

TABLE I

SUPERCONDUCTING ELEMENTS

Element	T_c°K	Element	T_c°K
Technetium	11.2	Aluminum	1.2
Niobium	8.0	Gallium	1.10
Lead	7.22	Rhenium	1.0
Vanadium	5.1	Zinc	0.91
Tantalum	4.4	Uranium	0.8
Lanthanum	4.37	Osmium	0.71
Mercury	4.15	Zirconium	0.70
Tin	3.73	Cadmium	0.56
Indium	3.37	Ruthenium	0.47
Thallium	2.38	Titanium	0.4
Thorium	1.39	Hafnium	0.35

TABLE II

SUPERCONDUCTING COMPOUNDS

Compound	T_c°K	Compound	T_c°K
Nb_3Sn	18	SnSb	3.9
NbN	14.7	Tl_2Hg_5	3.8
MoN	12	$PbTl_2$	3.8
Nbc	10.3	WC	3.8
ZrN	9.5	ZrB	3.0
TaC	9.4	W_2C	2.7
MoC	7.9	Mo_2C	2.7
Pb_5Na_2	7.2	Sn_2Au	2.7
Pb_2Au	7.0	Sn_4Au	2.7
Tl_3Bi_5	6.4	VN	2.4
NbB	6	ZrC	2.3
Tl_7Sb_2	5.2	Au_2Bi	1.7
Mo_2N	5	CuS	1.6
MoB	4.4	PbSb	1.5
TaSi	4.4	$CoSi_2$	1.27
Sn_3Sb_2	4.0		

trons and as the x, y, and z drives for memory arrays. Tantalum is a material that might be chosen when only its critical temperature (T_c=4.4°K) is considered, but even in its most purified form it has an efficiency of only 50 per cent, and usually only 10 per cent is obtained in practice because of absorbed gases and other impurities. Certain alloys appear to be promising because of the ease of adjusting the critical temperature and because of the possibility of obtaining a larger normal resistance and thereby increasing the speed of transition in a device. An example of a potentially useful alloy would be indium-lead. Other desirable properties such as insensitivity to alloy ratio and absorbed gases are also within the realm of possibility for a carefully chosen alloy.

Two-Hole Memory

Persistent-current memory cells for computer applications are well known and have been reported in the literature.[1–4] Essentially, the new cell consists of a set of two holes in a superconductor with a narrow superconducting bridge between them, as shown in Fig. 2.

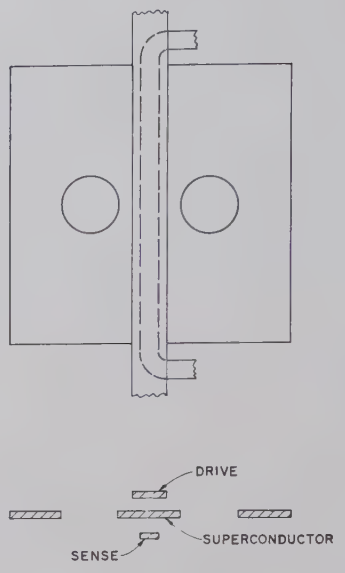


Fig. 2—Two-hole memory cell.

A drive wire placed above the bridge is able to induce and store a persistent current in the bridge and around the holes. The storage of persistent current is based on the principle of zero resistance and exclusion of time variation of magnetic field in a superconductor. This exclusion of time-varying magnetic field can be easily derived from Maxwell's equation

$$\text{Curl } \overline{E} = -\frac{\partial \overline{B}}{\partial t}, \qquad (1)$$

and from the fact that the electric field must be zero in a super-conductor. This yields

$$\frac{\partial \overline{B}}{\partial t} = 0. \qquad (2)$$

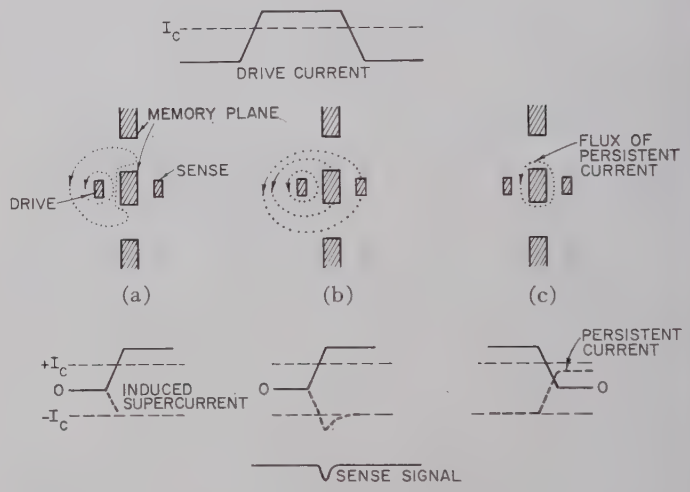


Fig. 3—Operation of two-hole memory cell.

The result is that the magnetic induction vector in a superconductor must be a constant with time (actually this constant turns out to be zero).

With this in mind, and with the help of Fig. 3, one can quickly review the principle of information storage in a cryoelectric memory cell.

Assume a drive current as shown by the solid curve at the top of Fig. 3(a) which shows that the field of the drive current cannot penetrate the bridge because the latter is in the superconducting state. The superconductor opposes the change of magnetic field by creating an induced current in the bridge and around the holes exactly equal and opposite to the drive current. However, there is a limit to how much field the superconductor can oppose, and when the field reaches a threshold value called the critical field (equivalent to a certain critical current represented by I_c in the drive wire) the superconductor switches the superconducting bridge to the normal state and links a sense wire placed directly underneath the bridge [Fig. 3(b)], thus inducing in the wire a voltage proportional to the slope of the drive signal while the latter is still rising. Fig. 3(c) shows what happens when the drive signal is removed. It is assumed that after the occurrences in Fig. 3(b), the bridge has returned to the superconducting state. Notice that a field exists around it now. In Fig. 3(c) the driving field is removed, but since the superconductor allows no change of field (the field around the bridge cannot collapse without cutting it) a persistent current will be set up to keep the field as it was before. It can be shown that the magnitude of the persistent current is approximately equal to the amount ΔI by which the drive cur-

[1] J. W. Crowe, "Trapped-flux superconducting memory," *IBM J. Res. & Dev.*, vol. 1, pp. 295–303; October, 1957.
[2] R. L. Garwin, "An analysis of the operation of a persistent-super current memory cell," *IBM J. Res. & Dev.*, vol. 1, pp. 304–308; October, 1957.
[3] E. H. Rhoderick, "Superconducting computer elements," *Brit. J. Appl. Phys.*, vol. 10, pp. 193–198; 1959.
[4] E. C. Crittenden, J. N. Cooper, and F. W. Schmidlin, "'The Persistor'—a superconducting memory element," Proc. IRE, vol. 48, pp. 1233–1246; July, 1960.

rent exceeds the critical value. A cross section of this cell is shown in Fig. 4, and an array of 6 cells using only one drive line per cell is illustrated in Fig. 5.

Memory cells of this type work very well, but only on an individual basis because wide variations between cells are exhibited when arranged in an array. This defect in uniformity is undesirable and the resulting lack in reproducibility does not allow the integration of such cells in large arrays in a large capacity memory system. The nonuniformity stems from the fact that one cannot get sharp edges around holes obtained by ordinary metal evaporation processes even under the most careful evaporation conditions. The metal, during the evaporation, diffuses under the edges of the evaporation masks and produces a layer with edges as shown in Fig. 6, while the desired condition on the edges is represented by the dashed lines in the figure. The result is that the high current density at the edges will make them switch to the normal state before the rest of the bridge and the heat generated by this action can be sufficient to raise the temperature of the material above the critical value and bring it entirely to the normal state. Slight variations in the shape of the edges cause the switching action to occur at widely different current levels and the result is a very severe lack of uniformity. To obviate such a drawback it is desirable either to clean up the edges or to remove them altogether. The first approach is feasible if one is dealing with few elements of large size. However, it is very inconvenient if the elements are numerous and of microscopic sizes, and the second approach, the elimination of the edges, is more practical.

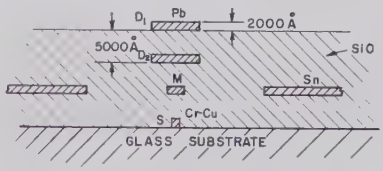


Fig. 4—Cross section of a persistent current memory cell.

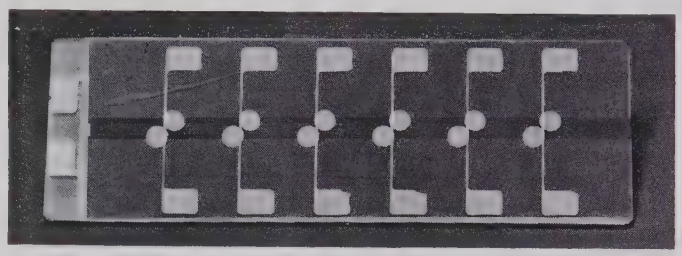


Fig. 5—Six-cell array on a 1-by-3-inch microscope slide.

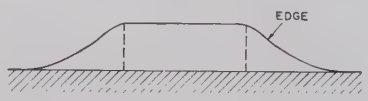


Fig. 6—Cross section of an evaporated film.

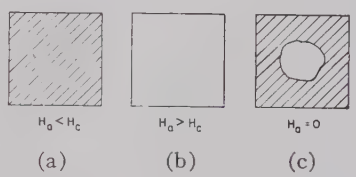


Fig. 7—Flux-trapping in a superconducting film (after Alers).

Continuous-Plane Memory

The edge effects are eliminated by using a continuous plane for the memory. The persistent current is stored in superconducting regions surrounding normal areas created in the material by the magnetic field of the drive current. The flux of the persistent current is trapped in the normal portions of the material. The possibility of trapping fluxes in a continuous superconducting film has been shown in the literature.[5] By using the Faraday effect (the rotation of the plane of polarization of a light beam when it is passed through a magnetized substance in the direction of the applied field) one can actually observe normal and superconducting portions of a sample under the application of a magnetic field, since the superconducting portion is opaque to magnetic fields. The observations made by Alers are summarized in Fig. 7. In this figure, H_c is the critical field of a lead sample and H_a is the applied field. The shaded portions represent superconducting areas and the nonshaded portions represent normal areas. It was found that as long as the applied field remained below the critical value the entire sample remained in the superconducting state [Fig. 7(a)]. For $H_a > H_c$ the sample was brought to the normal state [Fig. 7(b)] but a portion of the material around the center remained in the normal state after the removal of the applied field [Fig. 7(c)]. The implication of this result is that a portion of the applied flux remained trapped in the material. The existence of this trapped flux automatically implies the coexistence of a persistent current to sustain it. The pattern of Fig. 7(c) was reported to be "indefinitely stable when the field was held constant." Thus it is possible to store a persistent current in a continuous superconducting film by "punching" normal areas in it.

The magnitude of the trapped field H_s in relation to the persistent current density J_s can be calculated by considering the energy associated with the persistent current. Let the momentum per unit charge in the superconductor be $\overline{G}$, defined as

$$\overline{G}_L = \lambda \overline{J}_s, \quad (3)$$

where λ is a constant in [time]^2 related to the number n (number/unit volume), mass m, and charge e of the

[5] P. B. Alers, "Structure of the intermediate state in superconducting lead," *Phys. Rev.*, vol. 105, pp. 104–108; January, 1957.

[6] M. Von Laue, "Theory of Superconductivity," Academic Press, Inc., New York, N. Y., pp. 12–22; 1952.

super electrons by the relation

$$\lambda = \frac{M}{ne^2} \sec^2. \tag{4}$$

The above definition of $\overline{G}$ is in Lorentz units. In the MKS system one must replace λ by $\mu c^2\lambda$ where μ is the permeability of the material, and c is the velocity of light. Thus in the MKS system,

$$\overline{G} = \mu c^2 \lambda \overline{J}_s. \tag{5}$$

Also, according to London,[7]

$$\frac{\partial \overline{G}}{\partial t} = \frac{\partial}{\partial t}(\lambda \overline{J}_s) = \overline{E}. \tag{6}$$

In the MKS system,

$$\frac{\partial}{\partial t}(\mu c^2 \lambda \overline{J}_s) = \overline{E}. \tag{7}$$

The energy density is related to the electric- and magnetic-field vectors by the relation

$$\nabla \cdot [\overline{E} \times \overline{H}] = \overline{H} \cdot \nabla \times \overline{E} - \overline{E} \nabla \times \overline{H}. \tag{8}$$

This yields (considering the fact that there is no Joule heat loss in the superconductor)

$$\nabla \cdot [\overline{E} \times \overline{H}] + \frac{\partial}{\partial t}[1/2\epsilon E^2 + 1/2\mu H^2 + 1/2\mu c^2 \lambda J_s^2] = 0. \tag{9}$$

In a normal conductor there is a factor σE and the factor $\frac{1}{2}\mu c^2 \lambda J_s^2$ is missing. Its existence in a superconductor represents the energy associated with the persistent current. When a field H_s is trapped in the superconductor the energy associated with it is $\frac{1}{2}\mu H_s^2$. The equilibrium condition in a superconductor carrying a current I_s of density J_s and sustaining a field H_s is thus

$$1/2\mu H_s^2 = 1/2\mu c^2 \lambda J_s^2,$$

and the field associated with the persistent current is

$$H_s = cJ_s\sqrt{\lambda}. \tag{10}$$

The continuous sheet memory can be made by taking advantage of this flux trapping by using perpendicular strips of wire which carry the drive current over a thin superconducting plane to trap the flux. A sense winding is placed on the other side of the superconducting plane to detect the memory action (Fig. 8). In Fig. 8, the X and Y drive currents are sent in the direction shown by the arrows. The fields on the surface of the superconducting plane at the intersection of the drive conductors are represented by the dots and crosses near the intersection. They reinforce at the upper-left and lower-right sides of the intersection and cancel at the other sides. The result is a net concentration of flux at points A and

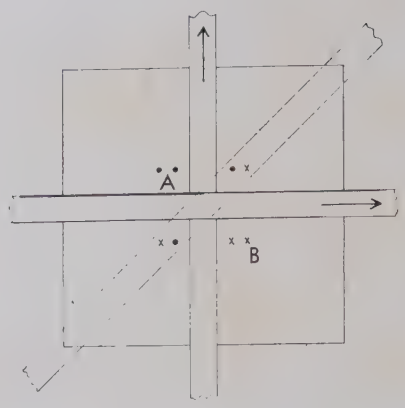


Fig. 8—Continuous-sheet memory cell.

B. The following discussion illustrates how this flux can store a persistent current in a superconducting plane. Fig. 9 shows a cross-section of the portion of the plane directly under the intersection of the drive wires at different times during the application of a drive pulse on the X and Y wires of the device. The cross hatched areas represent superconducting regions. The four parts represent the state of the device when: *a*) the drive current is below the critical value, *b*) the drive exceeds the critical value and the plane switches, *c*) the center portion of the plane becomes superconducting again, and *d*) the drive current is removed leaving a trapped flux in the plane and a persistent current to maintain it. The direction of the flux and of the persistent current can be changed by reversing the polarity of the drive currents. The storage of persistent currents in the plane constitutes a memory action. A positive current can be used to write into the memory and a negative current of the same amplitude to read (Fig. 10). The sense signal appears during the portion of the leading edge of the driving pulses starting when the drive currents reach the value of I_c. As will be shown later, the device is extremely fast.

For a given cell dimension it is possible to determine the magnitude of drive current required for storage and the approximate width of the persistent current path. The algebraic sum of the fields around the two drive wires give a net perpendicular field

$$H_p = 2H_{DP} \tag{11}$$

and a net tangential field

$$H_T = H_{DT}\sqrt{2} \tag{12}$$

at the surface of the superconducting memory plane (Fig. 11). The value of H_{DT} represents the field due to each drive current, and

$$H_{DT} = \frac{I_D}{W}, \tag{13}$$

where I_D is the drive current and W is the width of the drive wire. The memory will switch when $H_T = H_c$, the

[7] F. London, "Superfluids," vol. 1, John Wiley and Sons, Inc., New York, N. Y., p. 29; 1950.

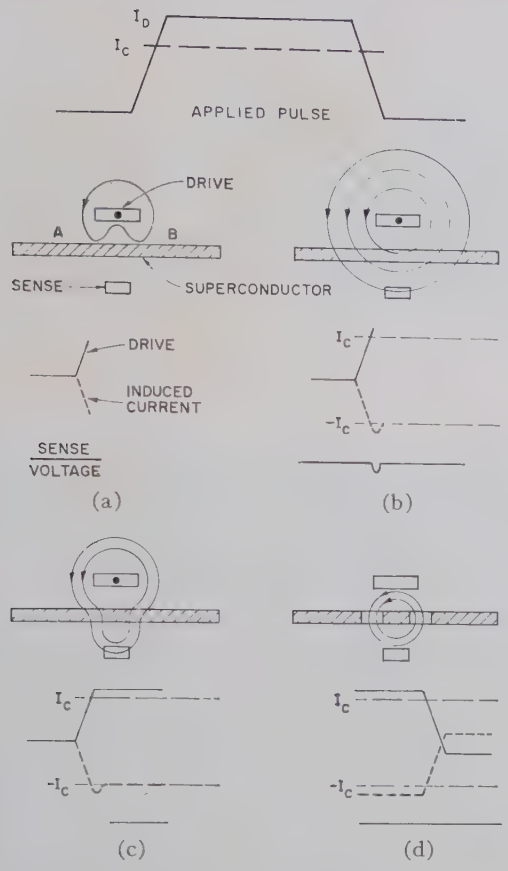


Fig. 9—Operation of a continuous-sheet memory cell.

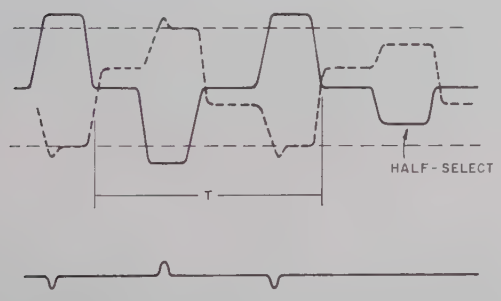


Fig. 10—Read-write cycle for a continuous-sheet memory.

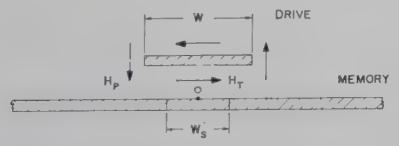


Fig. 11—Fields applied to memory plane.

critical value of the field at the surface of the superconductor. The critical drive current I_c is such that

$$I_c = \frac{1}{\sqrt{2}} W H_c. \tag{14}$$

A persistent current will be stored if $I_d > I_c$, and its magnitude I_s is such that

$$I_s = I_D - I_c. \tag{15}$$

The corresponding trapped flux H_s is

$$H_s = \frac{I_D\sqrt{2} - I_c\sqrt{2}}{W} = \frac{I_s\sqrt{2}}{W}. \tag{16}$$

This flux is related to the persistent current I_s by the energy equation (10) which can be rewritten as

$$H_s^2 = c^2\lambda J_s^2. \tag{17}$$

Assuming that the persistent I_s flows through a path of width W_s, one can write

$$\left(\frac{I_s}{2W_s}\right)^2 = c^2\lambda J_s^2. \tag{18}$$

The current density penetrates a thickness Z of superconductor such that[8]

$$J_s = \frac{1}{\sqrt{c^2\lambda}} H_s \epsilon^{-Z/\sqrt{c^2\lambda}}. \tag{19}$$

If the thickness of the superconductor is of the same order as the penetration depth $1/\sqrt{c^2\lambda}$ one gets

$$J_s \approx \frac{1}{\epsilon\sqrt{c^2\lambda}} H_s \quad \text{or} \quad c^2\lambda J_s^2 = \frac{H_s^2}{\epsilon^2}. \tag{20}$$

Substitution into (18) yields

$$\left(\frac{I_s}{2W_s}\right)^2 = \frac{H_s^2}{\epsilon^2}. \tag{21}$$

The value of H_s is obtained from (16), and when substituted into (21) gives

$$\left(\frac{I_s}{2W_s}\right)^2 = \left(\frac{I_s\sqrt{2}}{\epsilon W}\right)^2$$

or

$$W_s = \frac{\epsilon}{2\sqrt{2}} W. \tag{22}$$

Thus, the persistent current in the continuous sheet memory follows a path which is about 0.96 times the width of the drive line (Fig. 12).

The radius of the artificial holes, assumed for simplicity to be circular, can be determined from the flux linkage between them due to the persistent current. Once the bridge W_s returns to the superconducting state with I_s flowing, the sizes of the normal regions in Fig. 12 adjust themselves so that the trapped flux has the critical value H_c in these areas. Since the magnetic induction B is μH where μ is the permeability of the material, and since the total flux $\Delta\Phi$ linking the material is constant, one can write that the area $A = \pi r^2$ of the hole must be such that

$$\mu H_c \pi r^2 = \Delta\Phi. \tag{23}$$

[8] M. Von Laue, "Theory of Superconductivity," Academic Press, Inc., New York, N. Y., p. 26; 1952.

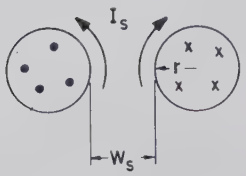


Fig. 12—Persistent current paths in the memory plane.

This flux cannot be obtained easily from the properties of the material, but it can be approximated from the magnitude E and duration τ of the sense voltage, assuming that the rise time of the drive signal is linear with time. Since the sense winding is actually very close to the memory plane and to the drive line, it is reasonable to assume that all of the flux H_s which has penetrated the superconducting plane has been able to link the sense winding. The integral of the sense voltage

$$e = \frac{d\phi}{dt}$$

yields under the previous assumptions

$$e\tau = \Delta\phi. \tag{24}$$

Thus, from (23), one can say

$$\mu H_c \pi r^2 = e\tau$$

and

$$r = \sqrt{\frac{e\tau}{\pi\mu H_c}}. \tag{25}$$

For a typical tin cell switching in 10^{-8} seconds with a sense voltage of 10^{-3} v operating at a temperature of 3.6°K at which $\mu H_c = 30$ gauss or 30×10^{-4} weber/m^2, the radius of the normal areas is about 3.3×10^{-5} meter of 1.3 mils. Thus a cell with a 10-mil drive would occupy a total space less than 16 mils wide. For a density of 10^9 cells per cubic foot or 10^4 per square inch the width of the drive line must be less than 5 mils.

Results

Successful results of tests on a 6×1 array of memory cells of the continuous-plane type lead to the design and construction of a 10×10 array. The 6×1 unit is shown in Fig. 13, and the 10×10 in Fig. 14. The array was made by vacuum deposition of the metallic and insulation layers, and the vacuum equipment is shown in Fig. 15. The metallic layers were 2000 to 3000 angstroms thick; the insulation layers were 5000 to 15,000 angstroms thick. The deposition was carried under a vacuum of 10^{-5} to 10^{-7} mm of mercury. The evaporation was carried through masks made by photoresist-etching techniques. With special care, lines 10^{-3} inch wide are obtainable. The material used for the memory plane is tin (critical temperature = 3.7°K). The drive and sense lines were vacuum-deposited lead. The lead is a superconductor at the temperature of operation (3.6°K)

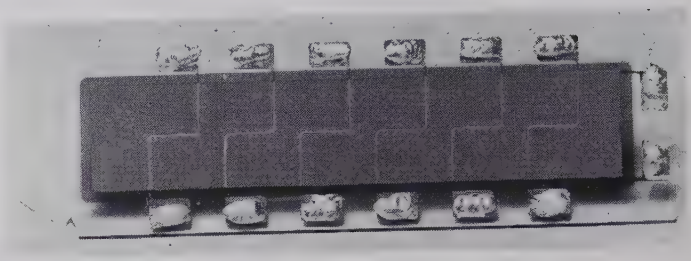


Fig. 13—6-by-1 continuous-sheet memory.

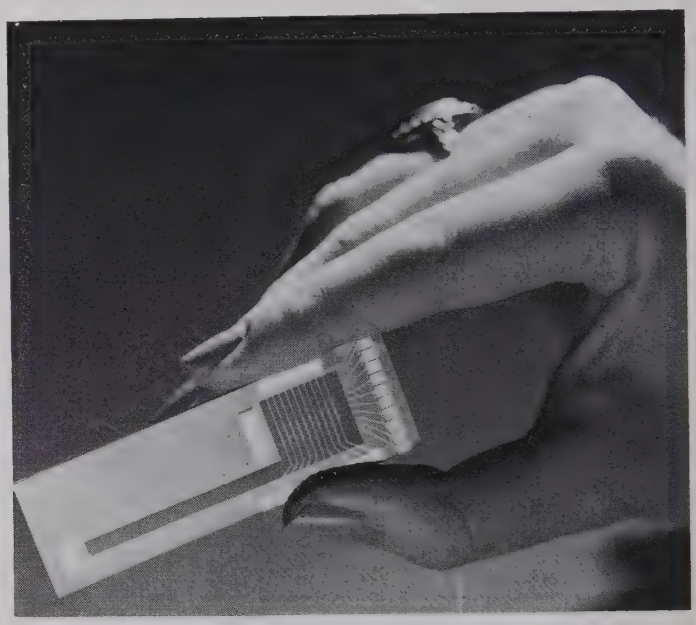


Fig. 14—10-by-10 continuous-sheet memory.

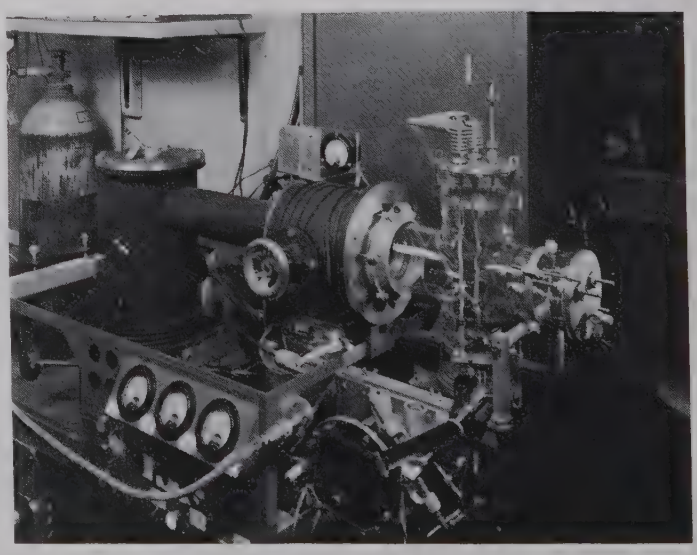


Fig. 15—Vacuum equipment used to fabricate memory planes.

and therefore consumes no power. The insulation used was silicon monoxide with special techniques used to obtain layers free of pin holes. The thickness of the layers is carefully and accurately controlled by a calibrated crystal oscillator whose frequency is varied according to the loading effect due to the thickness of the layers, the crystal being placed inside the vacuum system. For convenience, the width of the drive and sense lines were made to be 10 mils. This size is convenient for the present operations, but it is by no means a physical limit on the final size of our device. For the cells shown in Fig. 13, drive currents of magnitude equal to 300 ma were used successfully. For the sample shown in Fig. 14 values ranging from 30 to 80 ma were used according to the temperature of operation. The sense voltage varies from 0.8 to 8 mv, according to the rise time of the driving pulses which ranged from 200 to 20 nsec. An exploded view of the 10×10 memory array is illustrated in Fig. 16, and Fig. 17 represents the completed memory with the details of the drives, memory plane, and sense winding. A read-write cycle and the corresponding sense signal appear in Fig. 18. Three points are made evident by this illustration: the half-select drives alone cause no output signal, rewriting the same information causes no output signal, and the signal-to-disturb ratio is very large because the superconducting plane acts as an excellent shield between the drive lines and the sense line. As previously mentioned the device is quite fast. Fig. 19 shows the response of the continuous sheet memory to pulses ranging from 12 to 3 nsec with a rise time of 2 nsec. The fast write pulses were obtained from a mercury pulser. The read out was slower for the sake of convenience. Since the operation of the memory occurs within a portion of the rise time of the write pulse it is safe to say that the writing process occurred in a time duration of 1 nsec or less (an experiment is under way for a still faster response). The fact that a pulse of 3 nsec was sufficient to write in also proves that the thermal recovery time (if any) for the memory is less than 3 nsec. It may be worth mentioning that values reported in the literature for memory cells of the two-hole type are around 100 nsec.[1]

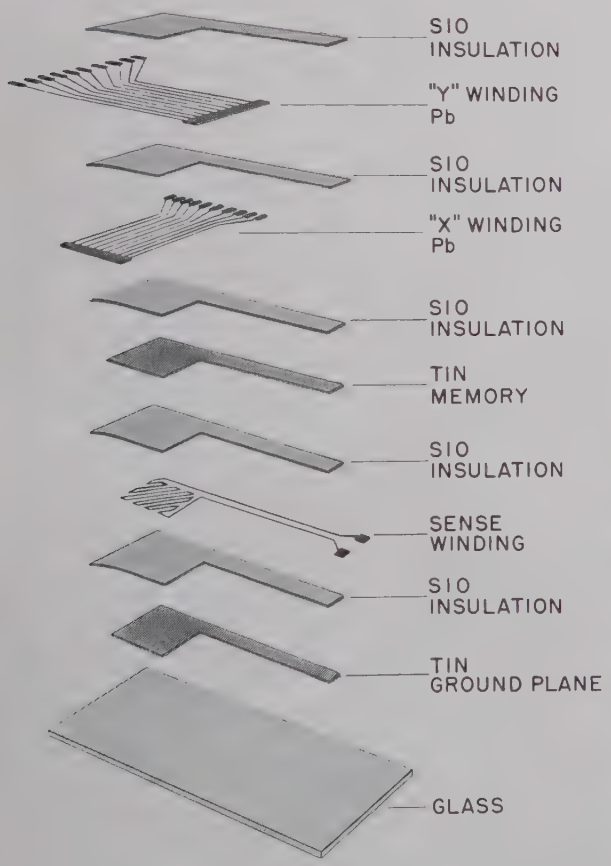


Fig. 16—Exploded view of 10-by-10 memory.

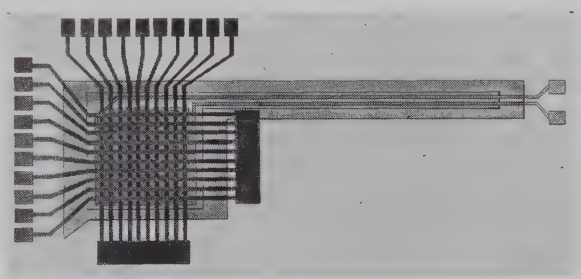


Fig. 17—Detail of construction: drive lines, memory plane, and sense line.

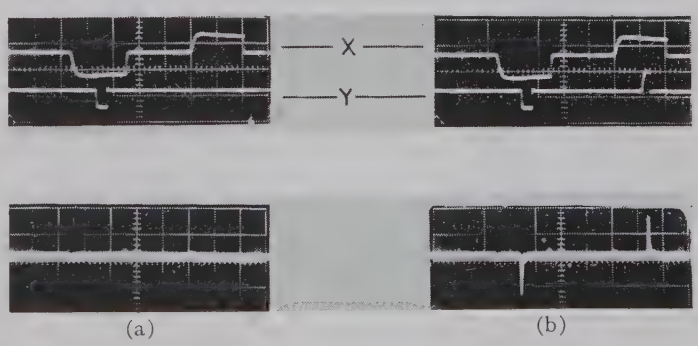


Fig. 18—Read-write cycle and sense signal. Horizontal scale: 1 μsec/div. Amplitude of drive pulses, 60 ma; amplitude of sense output, 0.8 mv. (a) Read-write cycle showing half-select drive and zero sense signal. (b) Full read-write cycle and corresponding sense output.

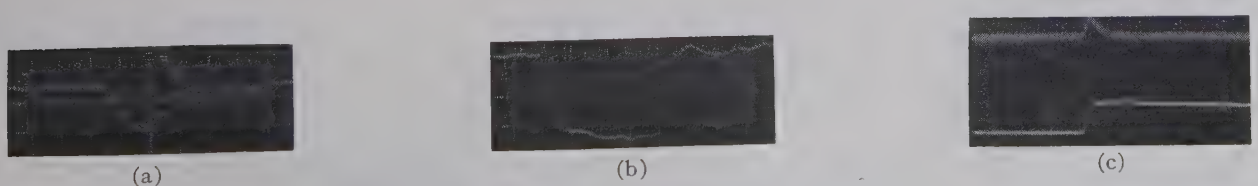


Fig. 19—(a) and (b) Write signal from mercury pulser. Horizontal scale: 2 nsec/div. (c) Sense signal and read pulse. Horizontal scale: 0.1 μsec/div.

Conclusion

The advantages of the continuous-plane memory are numerous. It is much easier to fabricate than the other types of superconducting memory because there is no need to make holes in the plane. Because of the elimination of the holes the reproducibility problems associated with the other types of cryoelectric memories are removed. Originally the continuous-plane memory cells could be made reproducible to within 7 to 10 per cent without special manufacturing techniques. With more special techniques it has been possible to obtain reproducibility to within $\frac{1}{4}$ of 1 per cent. The continuous-sheet memory is adaptable to X-Y-selection schemes and it exhibits no "delta" noise present in magnetic-core memories because of the shielding action of the superconducting plane. The cryoelectric memory can have a very high capacity. By spacing the drive wires 10^{-2} inches apart and by stacking 1000 planes evaporated on glass substrates 10^{-2} inches thick one gets a packing density of 10^{+7} per cubic inch.

The problems of addressing and sensing the memory should not be unduly difficult. Addressing can be done through a cryotron tree in which individual cryotrons need not have gain. Also possible is the use of semiconductor devices such as tunnel diodes or devices having special properties at low temperatures such as the cryosar. Sensing is greatly facilitated by the fact that the signal-to-noise ratio is inherently high, and no sophisticated strobing or integration methods are necessary. The problem is reduced to the amplification of a signal in the mv range. The speed of operation of the memory system depends chiefly on the addressing and sensing circuits, as the switching time of the element itself can be in the nsec range.

Acknowledgment

The authors wish to express their appreciation for the outstanding contributions made to this research effort by S. R. Hoffstein, C. M. LaValva, and L. R. Volpicelli.

Semipermanent Storage by Capacitive Coupling*

D. H. MACPHERSON†, senior member, ire, and R. K. YORK†, member, ire

Summary—**The need arises for reliable, economical high-speed, semipermanent stores for electronic-telephone switching systems, as well as for modern digital computers. A semipermanent or fixed store is one in which the stored information may not be changed by the machine that is able to consult it. These stores provide data security for such information as operational programs and test routines.**

A random-access store system where the memory elements consist of a matrix of printed capacitors has been developed. The store has a cycle time of 3 μsec and contains 1024 words each 34 bits long.

The access circuits developed enable one to utilize a matrix arrangement of components where a need exists for stores of thousands of words. These circuits consist of diodes and biased square-loop ferrite cores in the input, and magnetic gates, utilizing three transformers in a novel arrangement, in the output.

IN the quest for reliable high-speed digital systems a great need still exists for economical, reliable and changeable semipermanent stores which will provide data security for such information as operational programs and test routines. The present systems are limited in speed, where rotating magnetic drums are used, or rely on expensive and complicated error detection and correction circuitry where use is made of destructively read magnetic stores.

To overcome these shortcomings, two systems have been developed and recently reported: the flying spot store [1], which makes use of cathode-ray access to information stored on photographic emulsion and the permanent magnet twistor [2], in which information is stored by magnetically biasing the twistor [3].

This paper reports on a random-access store system utilizing a printed matrix of coupling capacitors. The store has a capacity of 1024 words, each 34 bits long, and operates in a 3-μsec cycle time. Although the specific system reported was developed to meet the needs of a stored-program-telephone switching system, these circuits can be used where larger and faster stores of this type are required. Laboratory tests indicate that the bit capacity can be increased by an order of magnitude.

Conventional circuits such as address registers, address translators and readout detectors will not be described.

The Memory

In the search for the semipermanent memory, matrix arrangements of resistors, diodes, transistors and capacitors were investigated. In the matrix, the horizontals are defined as the words and the verticals the bits of the word. If a "one" is to be stored, the component is placed in the associated bit position and, con-

* Received by the PGEC, October 24, 1960; revised manuscript received, March 6, 1961.

† Bell Telephone Labs., Inc., Murray Hill, N. J.

versely, if a zero is to be stored, the component is removed (not placed). To read the stored information a word at a time, a horizontal is pulsed while the verticals are sensed for the bit composition of the word.

Semiconductors were eliminated due to the relatively high failure rates as compared with resistors, capacitors and inductors [4]. The capacitor proved to be the most economical when fabricated in matrix arrays using well-known printed circuit techniques. The resulting memory is a "sandwich" of two conductor patterns separated by an insulating layer of mylar. The patterns consist of narrow conductors of copper to which are attached, at intervals along its length, larger areas of copper which serve as the capacitor plates.

In fabricating, a 1-ounce copper-clad board is etched to form the horizontal pattern of conductors and capacitor plates. A sheet of 2-mil mylar, with 1 mil of an adhesive, on both sides, is placed over this pattern. A 1-ounce copper sheet completes the sandwich. This unit is placed in a heated press to bind it into one unit. After photographic exposure, the uppermost copper is etched to form the vertical pattern. The conductors are 15 mils wide and the capacitor plates 125 mils in diameter. The bit capacitance, thus formed, is approximately 5 $\mu\mu$f. To aid in registration, the bottom capacitor plate is 25 mils larger in diameter. The matrix is shown schematically and photographically in Fig. 1 (next page). Although vertical and horizontal conductors are referred to, it can be seen in the photograph that all conductors are brought out to one side to provide a convenient means of connection. All bit positions are fabricated assuming the memory is to store all "ones." To store a "zero," the connection between the top capacitor plate and the conductor is severed.

Use of the Memory in a Store System

Matrix memories were limited to small sizes in the past as no economical circuit was available which provided the action of pulsing the selected word while at the same time holding all other inputs grounded. In this system the large size is made possible by the use of a biased square-loop ferrite-core switch, described in the next section. The need for this circuit can be illustrated by the use of Fig. 2. Assume that word A is selected and that capacitor C_{A1} is connected for a "one" and C_{A2} is disconnected for a "zero." When a pulse is directed to line A, it will be coupled to vertical 1 by capacitor C_{A1} but not coupled to vertical 2, as capacitor C_{A2} is assumed disconnected. If lines B and C were not grounded, however, the pulse coupled to vertical 1 would be coupled to vertical 2 by capacitors C_{B1} and C_{B2} as well as C_{C1} and C_{C2}.

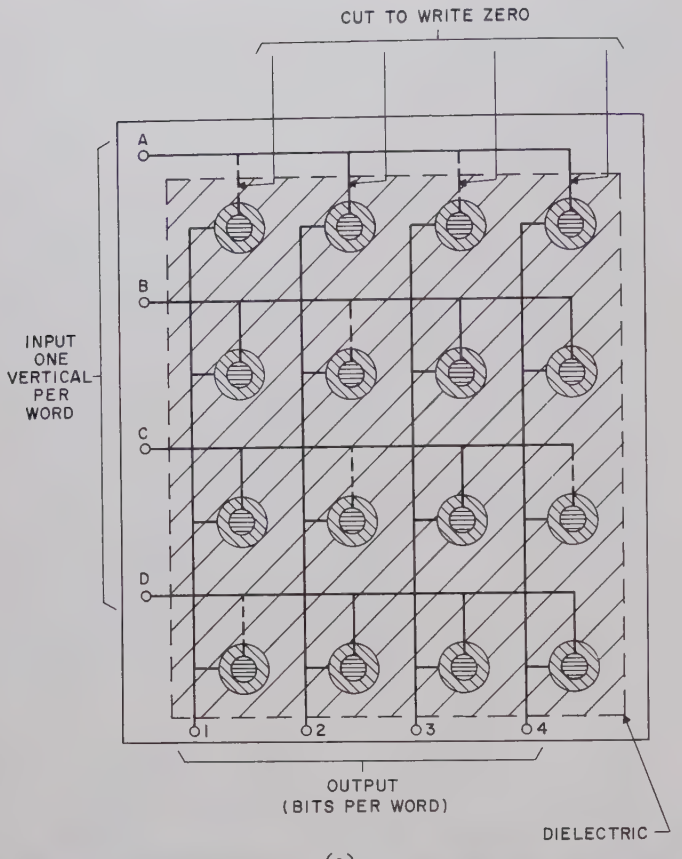
Additional circuitry must be added to the memory to insure an adequate SNR under the worst conditions. Referring to Fig. 2, it can be shown that the amplitude of the output signal is determined by a voltage divider consisting of the bit coupling capacitor of the selected word in series with the parallel combination of all other coupling capacitors defining the same bit of all other words in the memory. In the simplified equivalent circuit shown in Fig. 3, C_{A1} and C_{A2} are the bit-coupling capacitors of the selected word, while C_{T1} and C_{T2} are the parallel combinations referred to above. If the C_{A1} capacitor is connected for a "one" and all other capacitors on the same sense line are "ones," the output signal amplitude will be the same as the inverse case where the C_{A2} capacitor is disconnected and all other capacitors on the same sense line are disconnected. (The coupling between horizontal and vertical lines at zero locations must be considered). Although the example is an extreme case, a padding capacitor, C_{PAD}, is connected to each sense line to modify the voltage divider to achieve a one to zero ratio greater than unity. In the store described, the padding capacitor decreases the one output signal by approximately one half while the zero signal is decreased by an order of magnitude. The padding capacitor is etched on the wiring board in the same manner as the coupling capacitors. The padding capacitors are identified in Fig. 1 as the large areas at the end of the pattern.

The size of a memory of this type is limited by the sensitivity of the output signal detectors. Where very large word capacity is required, individual modules are provided which use common input access circuits and signal detectors. This method of operation requires that the biased-core switch select and pulse the same numbered word in each module and that a switch be provided to select and gate a word from one module to the detectors. This method of operation provides a coincident type of selection which is more economical than one word selection indicated earlier. The gate used in the store is described in a later section.

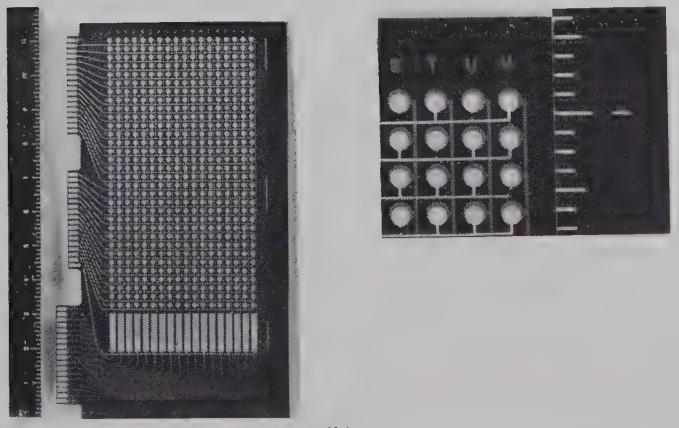
Input Access—Biased-Core Switch

The design of the biased-core switch has no resemblance to the typical switch used in magnetic-core stores and described in the literature [5]. These differences are that the switch must provide: 1) a short-duration voltage pulse, typically 0.5 μsec, and 2) a good ground connection to all memory words not selected.

The switch used is shown in Fig. 4. The cores are Cd-doped square-loop ferrite, and the diodes are WECo 2103. The bias holds the cores on the horizontal portion of the hysteresis loop to hold the inductance, looking into the output winding, to a minimum. Four turns are used to achieve a 20-volt signal output when a core is switched. This still provides the low impedance required when unselected. Selection of the core is by coincidence of a horizontal and vertical drive. The diodes provide the isolation required. If the typical biased core switch were used, where all cores on the selected horizontal and vertical-drive lines partially switch, the corresponding input lines to the memory would be raised above ground.



(a)



(b)

Fig. 1—(a) Capacitor matrix. (b) Photograph of capacitor matrix.

Gating One of *N* Outputs

A need for gating the output signals of the module to the common detectors is indicated above. In order to provide the gates necessary in an economical and reliable way, magnetic gating was selected over the more common methods using transistors or *p-n-p-n* diodes [4].

Fig. 5 illustrates, schematically, a single-magnetic gate or switch, composed of two transformers. In the normal or off state, any signal present at the input or primary windings induces equal and opposite voltages in the secondary windings resulting in no output signal. To perform the operation of gating, a pulse is applied to

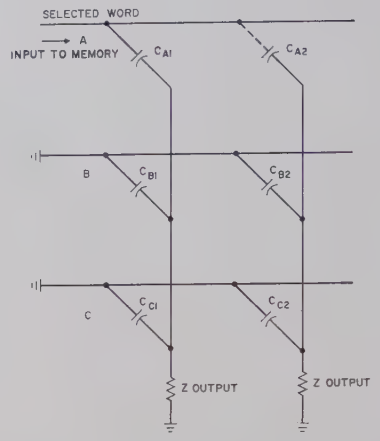


Fig. 2—Two-by-three capstor.

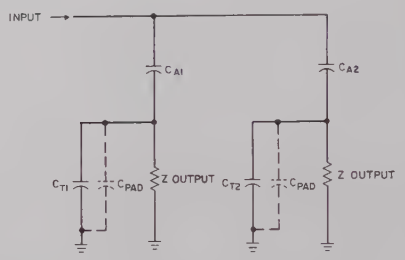


Fig. 3—Equivalent circuit.

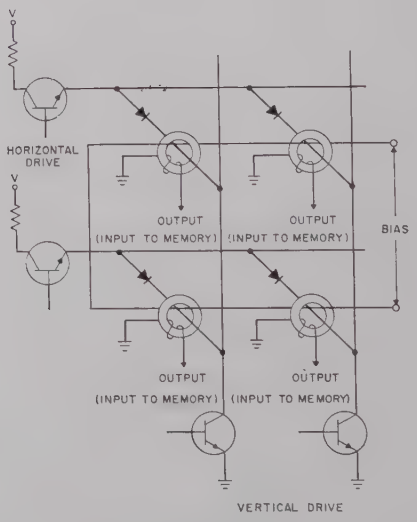


Fig. 4—Magnetic core access.

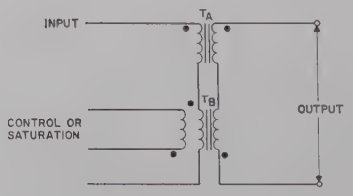


Fig. 5—Two-transformer gate.

the control winding such that transformer T_B is driven into saturation and held in that state while a signal is applied at the input. Since transformer T_B is now saturated, the input signal is induced into the secondary of transformer T_A only and appears at the output terminals. After gating the input signal, the control pulse is removed and transformer T_B assumes its normal operating point.

To increase the usefulness of this device, a means of forming a multi-input gate is needed. Some form of series or parallel connection of the output windings must be used. However, since the balance between secondary windings is not perfect, connecting the secondary windings in series for a common output results in the algebraic addition of the unbalanced voltages, limiting the number of series connected gates. If a parallel connection is used between secondary windings, the loading of these paralleled windings results in a high loss system.

In order to form a low-loss, multi-input device, using the single gate as a basic building block, a third transformer T_C is added. The configuration is shown in Fig. 6.

The secondary windings of the three transformers are connected in series and the secondary windings from each gate are connected in parallel. The input and control windings are shown for gate 1. The output winding is a series connection of the T_C transformers of each gate. The operation of a particular gate is as follows: We shall consider that an input is presented to gate 1. In the off condition where the control winding is not energized, two cancellations are present. The first is due to the secondary windings on transformers T_{A1} and T_{B1}, as described above. The second takes place between the secondary and output windings. This can be explained by considering a half cycle current flow in the secondary due to unbalance of transformers T_{A1} and T_{B1}. The assumed current flows down secondary windings 2 through N and up through the transformer T_{C1} secondary. The voltage induced in the output winding of transformer T_{C1} balances or cancels the sum of the voltages induced in the other output windings.

When the control pulse is applied to gate 1, transformers T_{B1} and T_{C1} saturate, eliminating both cancellations previously described. Under this condition, any signal directed to the input induces a voltage in the secondary winding of transformer T_{A1} only. The resulting secondary current induces a voltage in the output winding of gates 2 through N in the same manner as the leakage current described above. The output is the sum of these individual voltages.

The above configuration of transformers will hereafter be referred to as triple transformer gates (TTG).

A large transient voltage is induced in the output winding at the leading and trailing edges of the control current pulse. Since these transients, unlike the noise spike which immediately precedes the signal in magnetic stores, can be separated in time from the signal, no accurate strobing is required to identify the signal. The true signal is gated to the memory register while transformers T_{B1} and T_{C1} are fully saturated.

Although individual transformers were assumed in the previous discussion, they can be replaced by a single piece of ferrite. The N gates in schematic form in Fig. 6 are shown in Fig. 7 in the form of a multiaperture device without windings.

When using a sheet of ferrite to produce transformer cores, one encounters "crosstalk" between transformers whenever a signal or control pulse is applied. To isolate each "core" of the TTG, an isolation hole is cut between each active hole of a gate. Adjacent isolation holes are linked using a shorted turn as shown in Fig. 7. The shorted turn offers a high reluctance path to the flux, effectively isolating the active holes of each switch.

The cost of wiring may be considerably reduced by depositing single-turn secondary, output and shorted windings. Referring to Fig. 6, we note that the secondary windings are wired such that there are no connections external to the N gates and the output windings have only two external connections. In this application, the number of turns of the input winding is determined on an impedance matching basis and does not lend itself to the deposited form.

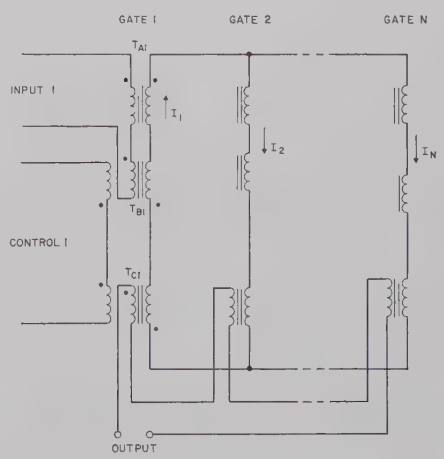


Fig. 6—Schematic for N triple-transformer gates.

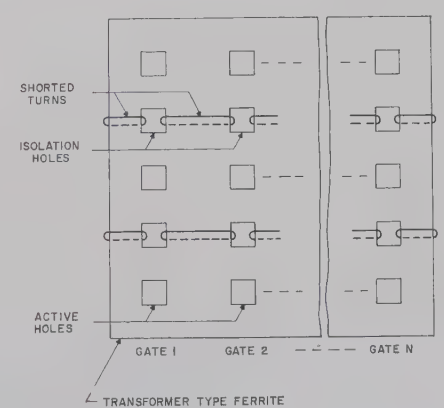


Fig. 7—Ferrite slab for N triple-transformer gates.

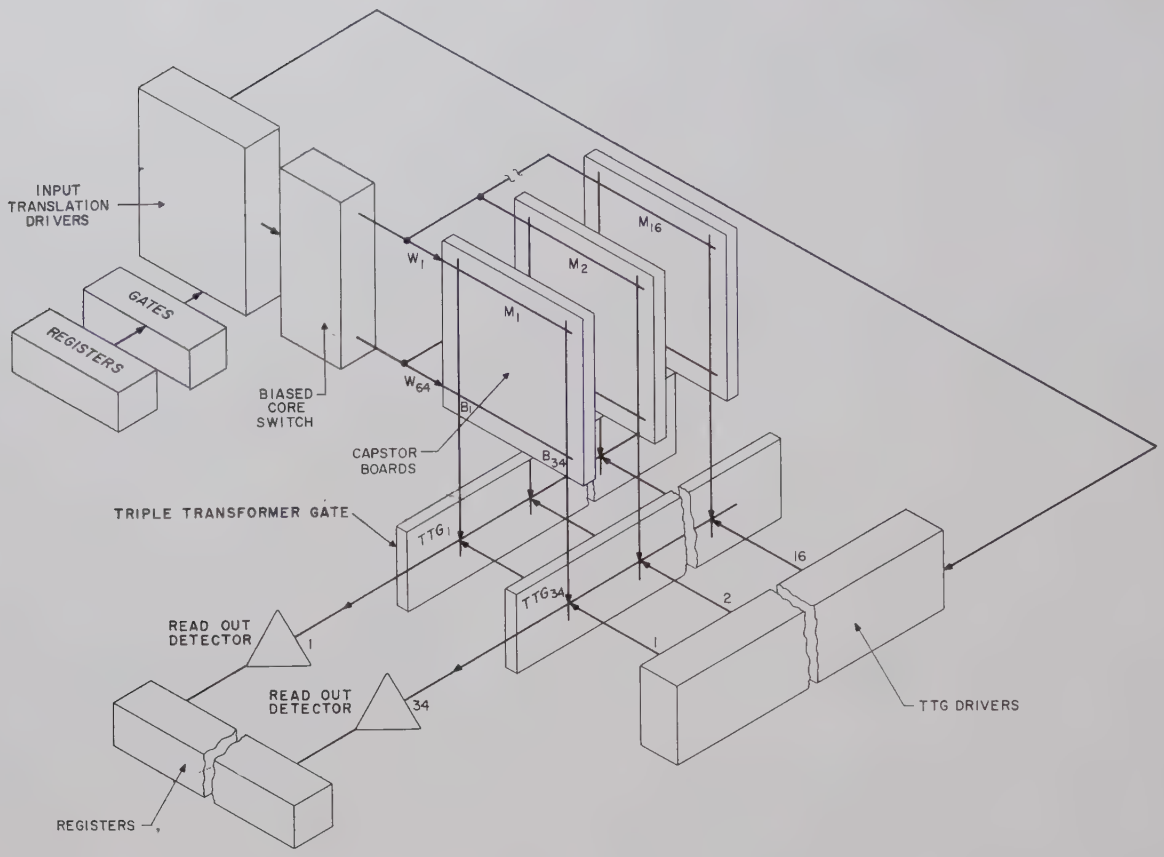


Fig. 8—Capstor semipermanent store system—1024 words, 34 bits.

The impedance matching between the gate and the output, in this system, is adequately provided by the inductance of the series connection of the single-turn, output windings. A larger number of gates will proportionately increase the internal impedance. Where a higher impedance is required, multiple turn output windings may be employed.

The Capstor Memory System

An operable capstor memory system of 1024 words, shown in block diagram form in Fig. 8, is composed of input drivers, capacitor modules, TTG gates, read-out detectors, and associated registers.

A module is a capstor board consisting of a matrix of capacitors where there are 64 inputs or words and 34 outputs or bits per word. The inputs to the modules are connected in parallel as described earlier. The module outputs are connected to the TTG. One TTG is required for each of 34 bits, each TTG containing 16 gates. The most significant bit of the words of each module is connected to TTG_1. The second most significant bit of each module is connected to TTG_2, etc. The output of each TTG is connected directly to the associated detectors. In order to gate the first module (34 outputs) to the detectors, gate 1 on each TTG (1 through 34) must be activated; therefore, a control or saturation winding is linked through these gates. One control winding is associated with each module (Fig. 8).

The input circuit of the capstor memory system consists of 64 biased core switches, one per input, driven by transistor circuitry. In order to select one of the 64 biased core switches, one out of eight horizontal and one out of eight vertical transistors are chosen. A similar transistor circuit is used to energize the control windings of the TTG's.

To clarify the operation and timing of the access circuitry, assume that the third word of the first module is to be interrogated. We shall also assume the binary address of the word has been translated to select transistors associated with the control winding of the TTG gates of module one. Approximately one μsec is required to turn on the magnetic gates, thereby providing a signal path from the output leads of module one to the detectors. While the magnetic gates are held in the "on" condition, the third biased-core switch of the input circuitry is pulsed. The pulse is coupled through the capacitors of module one, the magnetic gates, the detectors and finally sets the appropriate registers corresponding to the binary word selected.

Conclusion

The system described is suitable for the program store of electronic telephone switching systems or digital computers. To provide a reliable and economical system, use is made of printed circuit techniques to fabricate

the memory, and of TTG; coincident selection is employed to reduce the number of components in the access circuit, and magnetic circuits are employed where possible.

An experimental store using the techniques described has been designed, constructed and tested. In this system, the input to the capacitor modules was a 20-volt, 0.5-μsec pulse. The output of the store, for a one signal, was 10 mv across 50 ohms with a one to zero ratio of four to one.

Acknowledgment

The authors wish to thank A. R. Atwater for his help in fabricating the capacitor modules, J. A. Baldwin for assistance on the TTG, and R. D. Wiese for assistance on the over-all equipment design.

References

[1] T. S. Greenwood and R. E. Staehler, "A high-speed barrier grid store," *Bell Sys. Tech. J.*, vol. 37, pp. 1195–1220; September, 1958.
[2] J. J. DeBuske, J. Janik, Jr., and B. H. Simons, "A card changeable non-destructive readout twistor store," *Proc. Western Joint Computer Conf.*, San Francisco, Calif., March 3–5, 1959; pp. 41–46.
[3] A. H. Bobeck, "A new storage element suitable for large-sized memory arrays—the twistor," *Bell. Sys. Tech. J.*, vol. 36, pp. 1319–1340; November, 1957.
[4] J. H. Bollmann, "Component reliability," presented at the IRE Fall Lecture Series on Components and Reliability in Northern N. J.; October 20, 1959.
[5] J. P. Eckert, Jr., "A survey of digital computer memory systems," Proc. IRE, vol. 41, pp. 1393–1406; October, 1953.
[6] J. M. Wier, "A high-speed permanent storage device," IRE Trans. on Electronic Computers, vol. EC-4, pp. 16–20; March, 1955.
[7] R. D. Ryan, "A permanent high-speed store for use with digital computers," IRE Trans. on Electronic Computers, vol. EC-13, pp. 2–5; September, 1954.
[8] D. H. Looney, "New magnetic devices for digital computers," *Proc. 1960 Electronic Components Conf.*; May, 1960.

A Card-Changeable Permanent-Magnet-Twistor Memory of Large Capacity*

W. A. BARRETT†, F. B. HUMPHREY‡, member, ire, J. A. RUFF†, associate member, ire, and H. L. STADLER‖

Summary—**The card-changeable permanent-magnet-twistor memory is a large capacity (ca 10^5 bits) storage media for information that is infrequently changed. The information is stored in the form of small bar magnets bonded to a removable plastic card. The magnets, when magnetized, inhibit the switching of a section of twistor wire at a twistor-wire-solenoid crosspoint. For maximum information density the magnet shape and strength must be optimized with respect to the magnet's action on the inhibited crosspoint and the fringing action on neighboring crosspoints. The objective is a magnet with a small dipole moment, but with adequate inhibition of the twistor switching over a reasonable range of misposition. Suitable magnet shapes and a general discussion of the stray fields in a large array of magnets are given. For maximum capacity, the transmission characteristics of the twistor wire and the character of the access switch must be considered. Two novel structures of this memory permit increased information density and capacity. The feasibility of a random-access high-density memory submodule containing 360,000 bits in 0.7 foot[3] with a cycle time of 5 μsec has been demonstrated.**

Introduction

THE random-access memory of a modern digital computer is largely responsible for the machine's versatility. Nearly all general-purpose computers have only a *variable* memory in which the machine may both write and read information at high speed. However, special-purpose computers are now being built which also contain a *semipermanent* memory from which the machine may only read information at high speed. Such a memory might be used for program or constants storage where the information content is seldom changed. The semipermanent memory is further characterized by high reliability, low or zero volatility, and frequently very large capacity and low cost. Two examples of a semipermanent memory now in use are the flying-spot store[1] and the permanent-magnet-twistor memory.[2,3]

* Received by the PGEC, February 4, 1961. A portion of the work reported in this paper was done for the U. S. Dept. of Defense under Contract DA-30-069-ORD-1955.
† Bell Telephone Labs., Inc., Murray Hill, N. J.
‡ Jet Propulsion Labs., California Institute of Technology, Pasadena, Calif. Formerly at Bell Telephone Labs., Inc., Murray Hill, N. J.
‖ Ford Motor Co., Scientific Lab., Detroit, Mich. Formerly at Bell Telephone Labs., Inc., Murray Hill, N. J.

[1] C. W. Hoover, R. E. Staehler, and R. W. Ketchledge, "Fundamental concepts in the design of the flying spot store," *Bell Sys. Tech. J.*, vol. 37, pp. 1161–1194; September, 1958.
[2] D. H. Looney, "A twistor matrix memory for semipermanent information," *Proc. Western Joint Computer Conf.*, San Francisco, Calif., March 3–5, 1959; pp. 36–41.
[3] J. J. DeBuske, J. Janik, and B. H. Simons, "A card changeable nondestructive readout twistor store," *Proc. Western Joint Computer Conf.*, San Francisco, Calif., March 3–5, 1959; pp. 41–46.

The permanent-magnet-twistor memory utilizes an array of magnets bonded to removable cards as its information storage. The magnets are sensed by a solenoid-twistor structure. An early model of the memory, which contains 13,332 binary bits on 16 plastic cards, is shown in Fig. 1. A 16×32 biased core switch array[4] permits coincident-current address selection in 5 μsec.

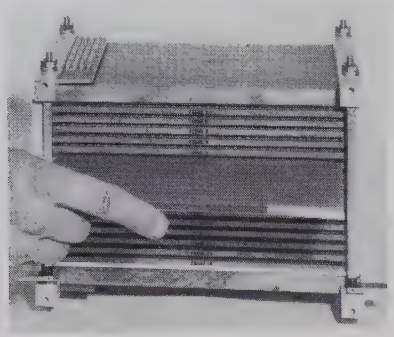


Fig. 1—Early permanent-magnet-twistor memory module of 512 addresses, 13,312 bits. Information-bearing magnets are fastened to the removable card.

Operating Principle

The principle of the memory can be best illustrated by reference to Fig. 2. A bit is formed by a single intersection of a twistor,[5] a copper wire wrapped with a 45° helix of permalloy ribbon especially processed to exhibit a square hysteresis loop, and a strip solenoid. The magnets are aligned with the twistor-solenoid structure. Two coincident currents I_1 and I_2 select a core in the biased-core switch and induce a current in the corresponding word solenoid. The field created within the solenoid is in a direction to reverse the magnetization of the twistors inside the solenoid. The magnetization of the permalloy is constrained to the helical direction. When the magnetization of that portion of twistor inside the selected word solenoid is reversed, a switching voltage will appear across the ends of each of the twistors. The appearance of this voltage is defined as a binary "one."

When a magnet is registered over the crosspoint, a much stronger static field is superimposed upon the field of the solenoid. The stronger field of the magnet inhibits magnetization reversal either by inhibiting the reversal (called Mode I) or by having previously reversed the bit (called Mode II). In either mode, little or no voltage is seen on a particular twistor if that magnet is present. Such a signal indicates a binary "zero." The interrogation is nondestructive. A number of magnets are mounted on a single card. The card's information may be altered by selectively magnetizing or demagnetizing the magnets to form "zeros" or "ones."

[4] J. A. Rajchman, "A myriabit magnetric-core matrix memory," Proc. IRE, vol. 41, pp. 1407–1421; October, 1953.

[5] A. H. Bobeck, "A new storage element suitable for large-sized memory arrays—the twistor," *Bell Sys. Tech. J.*, vol. 36, pp. 1319–1340; November, 1957.

Several design problems exist which are peculiar to this type of memory. The effects of the magnet's shape and strength determine the magnet-card-positioning tolerances relative to the associated twistor-wire solenoid crosspoint. Stray fields, *i.e.*, the field of neighboring magnets on a twistor crosspoint, essentially establish the maximum information density and certain other parameters. The maximum capacity of the memory is determined by the difficulty of sensing of the output signals. Finally, certain modifications in the memory structure permit increases in the information density. These situations will be discussed, and an example of a high-density module will be given.

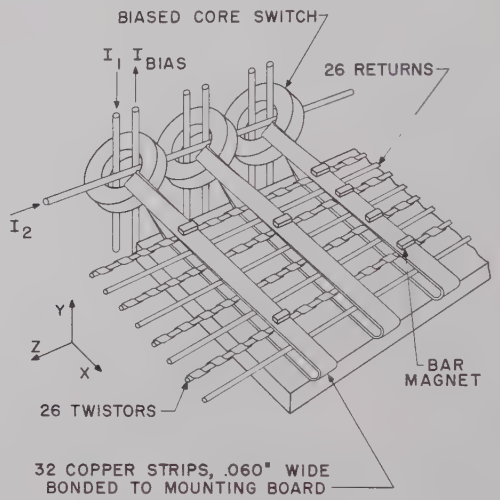


Fig. 2—Memory structure. I_1 and I_2 are access drive currents to core-selection switch. Presence or absence of a magnet over a twistor-strip solenoid crosspoint yields a "zero" or "one." Signals observed between twistor and return wire.

Permanent Magnet Design

Ideally, when a magnet is in position over the twistor bit, no flux change should occur in the twistor wire upon interrogation of that bit location. This condition should also persist if the magnet were mispositioned to some extent. A suitable magnet should therefore have a large field acting on the selected bit within this range and a zero field everywhere else. Such a rectangular magnetic field does not exist for a simple bar magnet, but may be approached with other magnet shapes. The variation in twistor signal when the magnet is moved perpendicular to the twistor axis will be examined first, and then the signal variation when the magnet is moved along the twistor axis will be considered.

Since the magnitude of the "zero" signal depends not only on the magnet's field, but on the bit in its magnetic environment, a memory plane as illustrated in Fig. 2 was used for this study. The entire twistor was premagnetized with a current of 250 ma. Following this, the switch core was pulsed and reset to yield a field of about 12 oe for at least 1.5 μsec in the strip solenoid. The output voltage was observed when the drive field opposed this premagnetization. The output voltage on

the twistor was measured at a fixed time (1.4 μsec) after the onset of the drive pulse. The chosen observation time is the optimum strobe time for the module of Fig. 1, and is not a significant aspect of the experiment. The twistor had a 0.003-inch diameter copper-core wire, wrapped with a specially processed 4-79 moly-permalloy tape 0.0003 inch×0.003 inch. The twistor flux per turn was about 8×10^{-10} webers, and its coercive force was about 4 oe. The direction of the drive field is defined as negative. All the data was taken by moving the magnet in a plane parallel to the solenoid-twistor plane. At all times the magnet was about 0.010 inch above the center of the solenoid-twistor intersection. The magnets were made of vicalloy, and were 0.001 inch thick and heat treated to obtain a remanent magnetization of 6000 gauss and a coercive force of 200 oe.

Consider first the effect of moving a simple bar magnet (0.030 inch wide×0.060 inch long) perpendicular to the twistor. The induced twistor voltage as a function of perpendicular magnet position is shown in Fig. 3. The magnet was oriented parallel to the twistor-solenoid plane with the long dimension parallel to the twistor axis. Mode I in Fig. 3 is for a magnet whose field is positive at the twistor, and Mode II is for a magnet whose field is negative at the twistor. The symmetry in Figs. 3–6 was always verified. Thus data were often taken for one side only and reflected for completeness.

The results of Fig. 3, Mode I, do not lend themselves to detailed interpretation. It can be seen that switching of the twistor is inhibited when the magnet is perfectly aligned. As the magnet is moved off center, however, an output is observed surprisingly soon, considering the 30-mil width of the magnet. This indicates that the magnet's field is not uniform laterally. Finally, a combination of higher effective drive and more available twistor flux increases the voltage output as the magnet is moved further off center.

Mode II (the solid curve in Fig. 3) operation is better understood. As the magnet is moved toward the twistor from some distance away, the static field at first increases the effective drive and hence increases the voltage output. Eventually the static field becomes large enough to switch the twistor bit of its own accord, and then the amount of flux available for switching by the solenoid field is greatly reduced. The magnitude of the zero when the magnet is correctly aligned is generally less than for Mode I, indicating that the top of the twistor hysteresis loop is not linear in this range.

As mentioned previously, it is desirable to have a reasonable tolerance for the positioning of the magnet to facilitate fabrication of the unit and to simplify insertion of the card. For example, if the memory is operating in Mode I, the zero output would be 0.6 mv with a perpendicular positioning tolerance of ±0.010 inch or 0.9 mv for a tolerance of about ±0.015 inch (see Fig. 3). The lower zero of 0.6 mv can be sacrificed to increase the positioning tolerance.

A more uniform field distribution can be obtained by shaping the magnet. Fig. 4 illustrates the improvement obtained by using a double-bar magnet. Here each bar is 0.015 inch×0.060 inch, so that the two bars together comprise the same size magnet as the solid bar of Fig. 3. The two bars are spaced 0.010 inch. The Mode I data of Fig. 3 are reproduced in Fig. 4 (dashed line) for comparison. The tolerance is increased from ±0.015 inch to ±0.030 inch (for a nominal zero of 0.9 mv). The sizes and spacing involved here illustrate the use of this technique to increase the mechanical tolerance in the transverse direction without increasing the strength of the magnet.

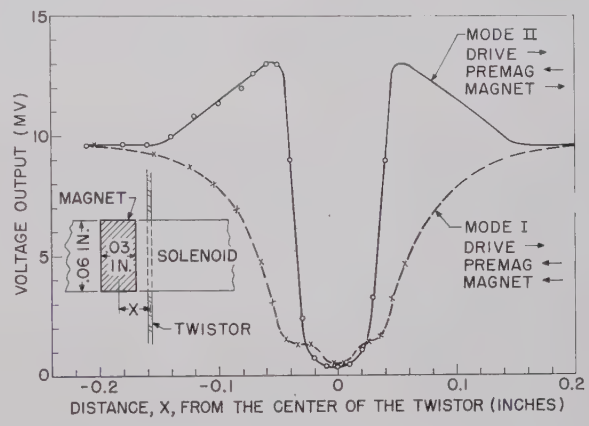


Fig. 3—The induced voltage between a twistor and return as a magnet of the size indicated is moved across the twistor. Arrows refer to relative directions of the interrogate field, twistor premagnetization and magnet field.

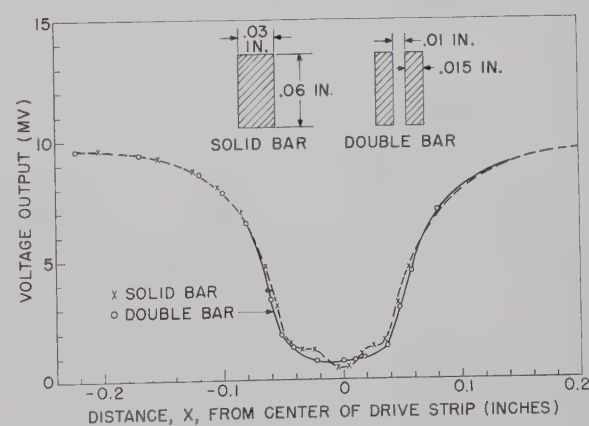


Fig. 4—The induced voltage between a twistor and return as a single- and a double-bar magnet of the size indicated is moved across the twistor. The direction of the various fields is for Mode I operation; the solid-bar curve is the same as the dashed curve in Fig. 3.

For the case of longitudinal alignment, the induced voltage as a function of position along the twistor is as shown in Fig. 5. The magnet is a single bar 0.030 inch ×0.060 inch as before. Its length is the same as the solenoid width. Clearly the magnetic field does not vary as rapidly with distance as in the transverse case. Less variation is expected because both the twistor bit and

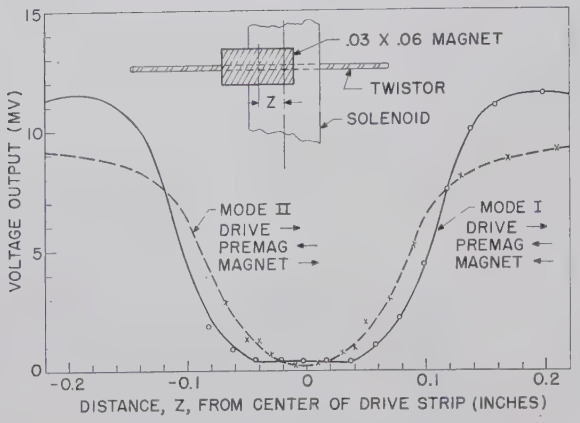


Fig. 5—The induced voltage between a twistor and return as a magnet of the size indicated is moved along the twistor. Arrows refer to the relative direction of the interrogate field, twistor premagnetization and magnet field.

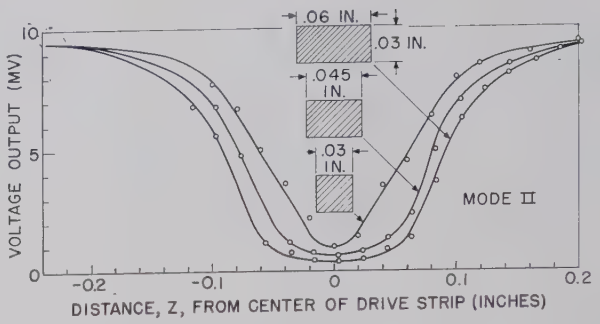


Fig. 6—The induced voltage between a twistor and return as various magnets of the size indicated are moved along the twistor. The direction of the various fields is for Mode II operation.

magnet are long in this direction. If the magnet had an ideally rectangular magnetic field, a roughly triangular pattern would be expected in which the base of the triangle is the sum of the magnet length and the solenoid width. The curves for Mode I and Mode II for the longitudinal case (Fig. 5) have the same general character as in the transverse case (Fig. 3), except that Mode I in the former resembles Mode II in the latter. Again, this difference is expected, since the field direction beyond the ends of the magnet is opposite to the field direction along the sides of the magnet.

Although multiple bars could be used for localized field shaping, a simpler variation is to shorten the magnet length. In Fig. 6, a magnet 0.030 inch wide has been shortened from 0.060 inch to 0.045 inch and then to 0.030 inch. The induced voltage in the twistor as a function of position is plotted for each case. The width of the solenoid remained at 0.060 inch. Clearly, any advantage gained through shortening the bar must require the acceptance of larger "zeros." Even so, the flexibility available for length changes is limited by the demagnetization of the magnet. It is desirable that the permanent-magnet material have a reasonably square hysteresis loop and a high coercive field to minimize this demagnetization.

To obtain magnet positioning tolerance in the transverse direction, a double-bar magnet is preferred to a single bar of the same strength. Further, the magnet may be shorter than the solenoid width. A shorter magnet permits an increased bit density, while still maintaining an adequate longitudinal positioning tolerance with a slight increase in the zero signal level.

Stray-Field Effects

Any magnetic field at a bit location other than the controlling magnet's field will alter the switching behavior of that location. The switching voltage of a "one" (no magnet) is more sensitive to stray fields than that of a "zero" (magnet in place). Since the effect of fields external to the memory may be eliminated by magnetic shielding, only the fields of other magnets in the memory need be considered. The largest possible fields will be considered first, and then the likelihood of weaker fields that can arise from the magnet array will be estimated.

When a magnet is far enough away from the origin, its field at the origin is approximated by the dipole equation

$$\boldsymbol{H} = -\operatorname{grad} \frac{\boldsymbol{M}\cdot\boldsymbol{r}}{r^3} \text{ (emu)}, \tag{1}$$

where $\boldsymbol{M}$ is the dipole moment. $\boldsymbol{M}$ is most conveniently defined in terms of the magnetization $\boldsymbol{I}=(\boldsymbol{B}-\boldsymbol{H})/4\pi$:

$$\boldsymbol{M} = \int\int\int_V \boldsymbol{I}dV. \tag{2}$$

The integral extends over the magnet volume V. An upper limit on $\boldsymbol{M}$ is obtained by assuming that $\boldsymbol{I}$ is everywhere equal to the remanent magnetization I_r and directed in the z direction. Thus

$$\boldsymbol{M} \leq I_r V k. \tag{3}$$

For a nonellipsoidal-shaped magnet, the dipole moment is best determined experimentally. A single-field measurement at a suitable distance from the magnet suffices to calculate $\boldsymbol{M}$ from (1). The dipole moments of the magnets used for the data of the preceding section were measured with a Hall-effect magnetometer. In Table I the experimental moments are compared with the upper-bound moments calculated with (3). The remanent flux density $B_r=5000$ gauss is a bulk measurement. The discrepancy between the measured and calculated upper bound is attributed to the magnet shape, indicating that about 60 per cent of the magnet is effectively magnetized.

TABLE I

DIPOLE MOMENTS OF VARIOUS VICALLOY MAGNETS, MEASURED AND CALCULATED UNDER ASSUMPTION OF UNIFORM MAGNETIZATION

Magnet All .001 inch thick Vicalloy II	Dipole Moment M (oe-cm^3)	
	Measured	Calculated (Upper Bound) $B_r = 5000$ gauss
1) 0.030 inch × 0.065 inch	7.2×10^{-3}	12.8×10^{-3}
2) 0.030 inch × 0.045 inch	5.2	8.9
3) 0.030 inch × 0.030 inch	3.1	5.9
4) 0.015 inch, 0.010 inch, 0.015 inch × 0.065 inch	6.9	12.8

The dipole field equation (1) may be written out in spherical coordinates (r, θ, ϕ), where $\boldsymbol{M}$ is directed along the z axis ($\theta = 0$). Thus

$$H_z = \frac{M}{r^3}(3\cos^2\theta - 1) \tag{4}$$

is the field of a single magnet located at (r, θ).

Eq. (4) states that if the magnet lies within either of the two cones bounded by $0 \leq \theta \leq 55°$ or $125° \leq \theta \leq 180°$, the magnet's z field will be positive, while if the magnet lies outside these cones ($55° \leq \theta \leq 125°$), its field will be negative. Therefore, the largest possible positive field of an array of magnets is obtained by adding the fields due to magnets inside the cones. The largest possible negative field is obtained from those magnets outside the cones. Eq. (4) implies that the field sum over the conical-magnet array increases logarithmically with the array size. This follows by integrating (4) over the volume bounded by the cones described above and the concentric spheres of radii r_1 and r_2. Since the volume element is $2\pi r^2 \sin\theta dr d\theta$, the integral contains a log r_2 term. The magnets are assumed to be uniformly distributed within the volume. On the other hand, the integral over the entire array between the spheres of radii r_1 and r_2 vanishes. Thus, when all magnets are in place, only a small stray field is found.

An accurate field calculation evidently requires a summation of (4) for a Cartesian array. Let x_0, y_0, and z_0 be the array spacings in the x, y, and z directions of Fig. 2. Also let the field point (the origin) be at the center of an array bounded by the planes $x = \pm X$, $y = \pm Y$ and $z = \pm Z$. Clearly, the largest positive and negative field sums H^+ and H^- will depend on all six of these parameters, as well as upon M. However, M and z_0 may be extracted as scale factors. Also, since H^+ and H^- depend only weakly upon the array size, X, Y and Z will be fixed. In all the calculations, $X = 16z_0$, $Y = 16z_0$ and $Z = 30z_0$. These are approximately the proportions of the memory in Fig. 1.

The largest possible field sums are given in Figs. 7–10 (next page) in the form of contours of constant $H^{\pm} z_0^3/M$ in the $(x_0/z_0, y_0/z_0)$ plane. The magnet orientation is + on the $y = 0$ plane, − at $y = \pm y_0$, etc., alternating from plane to plane for Figs. 7 and 8. This situation arises from folding a single twistor back and forth through successive planes. The largest possible positive field is in Fig. 7, and the largest (in magnitude) possible negative field is in Fig. 8. Figs. 9 and 10 are similar, except that plane pairs are alternated in magnet direction (++ −− ++ −− etc.).

As an example, consider the memory of Fig. 1, with magnets of type (1) in Table I. Here $x_0 = 0.1$ inch, $y_0 = 0.25$ inch and $z_0 = 0.25$ inch. Then Fig. 7 shows that $H^+ z_0^3/M = 50$. With $M = 7.2\times10^{-3}$ oe cm^3 (Table I), $H^+ = 1.4$ oe. Similarly, $H^- = -2.0$ oe from Fig. 8. The largest possible fields due to the six nearest neighboring magnets are $H^+ = 0.17$ oe and $H^- = -0.88$ oe. Consequently, most of the largest possible field arises from magnets beyond the six nearest neighboring magnets.

Several general remarks may be made about Figs. 7–10. Straight lines of constant bit density in any of the figures slope 45° to the left. It is clear that an optimum spacing ratio exists for any given bit density (z_0 scale) and appears to be close to $x_0 = y_0$. However, the contour for an equal positive and negative field (the dotted line) diverges from the line $x_0 = y_0$ appreciably.

A single-plane ($y = 0$) field summation is given in Fig. 11, where $H^{\pm} z_0^3/M$ is plotted as a function of x_0/z_0 and a constant array size ($16z_0$, $30z_0$). Here the positive field quickly approaches a limit determined by the end-on neighbors as x_0/z_0 becomes large. The negative field goes to 0 for large x_0/z_0. There is also a crossover at $x_0/z_0 = 0.6$, which is reflected in the limit $y_0/z_0 \to \infty$ in Figs. 7–10.

The preceding discussion is based wholly on the dipole-field approximation. The error involved is negligible at a distance large compared to the magnet length. However, the field cannot be accurately calculated close to the magnet, since the poles of the real magnet are distributed in an unknown manner. A second approximation may be made by assuming that the poles $\pm m$ are concentrated at each end of a bar of length l; then $M = ml$. The calculated field of this idealized magnet is given in Fig. 12 in comparison with the calculated dipole field. The field H_z for $\theta = 0$ (end-on) and $\theta = 90°$ (lateral) is normalized to M/r^3 and plotted. It is believed that the field of most real magnets lies between

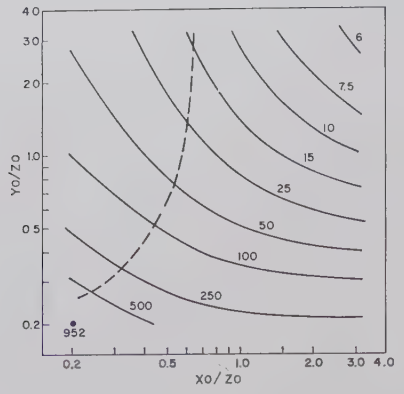


Fig. 7—Contours of the largest possible *positive* normalized z field $H_z{}^+z_0{}^3/M$ for a rectangular array of size $\pm 16z_0$ by $\pm 16z_0$ by $\pm 30z_0$, with magnets of moment M as a function of relative spacings x_0/z_0, y_0/z_0. Magnet direction alternates in the y direction, + for $y=0$, − for $y=\pm y_0$, etc.

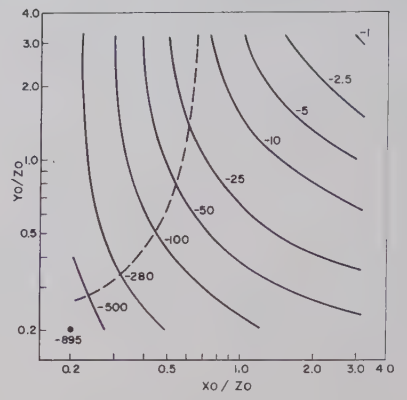


Fig. 8—Contours of the largest possible *negative* normalized z field $H_z{}^-z_0{}^3/M$ for a rectangular array of size $\pm 16z_0$ by $\pm 16z_0$ by $\pm 30z_0$, with magnets of moment M as a function of relative spacing x_0/z_0, y_0/z_0. The magnet direction alternates in the y direction, + for $y=0$, − for $y=\pm y_0$, etc.

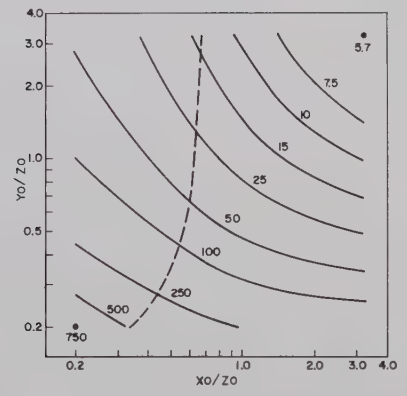


Fig. 9—Contours of the largest possible *positive* normalized z field $H_z{}^+z_0{}^3/M$ for a rectangular array of size $\pm 16z_0$ by $\pm 16z_0$ by $\pm 30z_0$, with magnets of moment M as a function of relative spacing x_0/z_0, y_0/z_0. The magnet direction alternates in pairs of planes; + for $y=0$ and y_0, − for $y=2y_0$ and $3y_0$, etc.

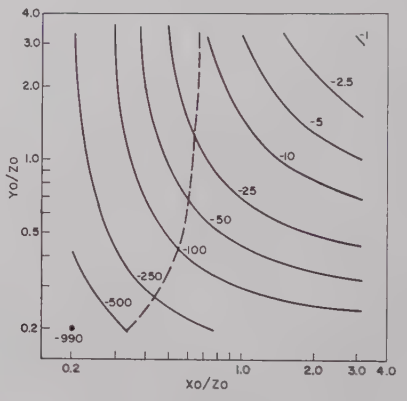


Fig. 10—Contours of the largest possible *negative* normalized z field $H_z{}^-z_0{}^3/M$ for a rectangular array of size $\pm 16z_0$ by $\pm 16z_0$ by $\pm 30z_0$, with magnets of moment M as a function of the relative spacing x_0/z_0, y_0/z_0. The magnet direction alternates in pairs of planes; + for $y=0$ and y_0, − for $y=2y_0$ and $3y_0$, etc.

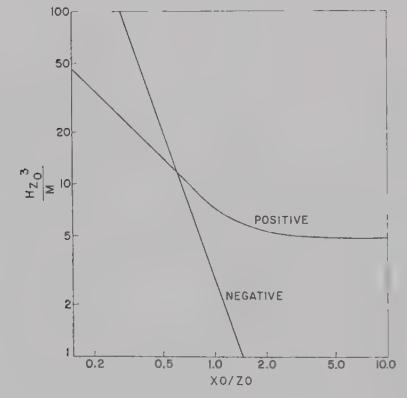


Fig. 11—Positive and negative largest possible normalized fields $Hz_0{}^3/M$ for a single plane ($y=0$) of magnets.

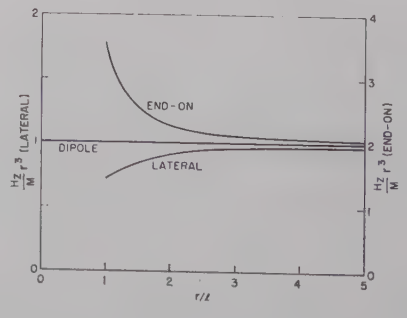


Fig. 12—Comparison of dipole z field to two-pole field of same moment M as a function of distance r. End-on curve and scale at right is for field along magnet axis. Lateral curve and scale on left is for z field along normal direction to magnet.

the two-pole and the dipole field. A field measurement of magnet (1), Table I, for example, has shown that the dipole model is valid within experimental error (± 0.2 oe) for $r/l > 2$.

Stray-Field Probability

The probability of obtaining a particular long-range field through various patterns of information storage should also be considered. The magnet patterns yielding the largest possible stray fields have been considered thus far. Even though such patterns are extremely unlikely to occur, the probability of a bit error due to the stray field should be evaluated. The *stray-field distribution* $p(H)$ at a location in the array may be defined by the probability $p(H)\Delta H$ that a field between H and $H+\Delta H$ is found at that location. In the absence of detailed information concerning the program or special codes stored in the memory, a random equally-weighted distribution of "ones" and "zeros" will be assumed. Then it may be shown that the mean (μ) and the standard deviation (σ) of the field distribution is given by (5) and (6):

$$\mu = \tfrac{1}{2} \sum H_i, \tag{5}$$

$$\sigma^2 = \tfrac{1}{4} \sum H_i^2. \tag{6}$$

Here H_i is the z component of field at the origin due to the ith magnet, and the sum extends over all magnets in the array. Neither μ nor σ depends on X, Y or Z, if enough magnets are included in the array. Both μ and σ are in field units (oe). The mean is the average field expected. The probability that the field, at some location, is greater than μ is equal to the probability that the field is less than μ. The mean is also half the algebraic sum of the largest possible fields H^+ and H^- previously defined. It may therefore be calculated from Figs. 7–10. The standard deviation is a measure of the distribution width. If the field distribution is Gaussian (as we expect it to be, approximately), the field will lie between $\mu+\sigma$ and $\mu-\sigma$ with a probability of $\frac{2}{3}$. Note that σ does not depend on the plane-to-plane organization of the magnets (since the sign of H_i is lost). Fig. 13 is a contour plot of $\sigma z_0^3/M$ similar to Figs. 7–10, previously described.

Clearly, the field distribution $p(H)$ is not simple. However, it is well approximated within one or two σ's from μ by the Gaussian distribution. In the distribution "tails" the Gaussian is an upper bound on the field distribution. The failure rate may therefore be roughly estimated through the use of the known Gaussian distribution, which is fully characterized by the two parameters μ and σ.

To find the failure rate, define h by (7):

$$h = \frac{|H_f - \mu|}{\sigma}, \tag{7}$$

where H_f is some stray field, beyond which a bit failure is expected under average memory operating conditions.

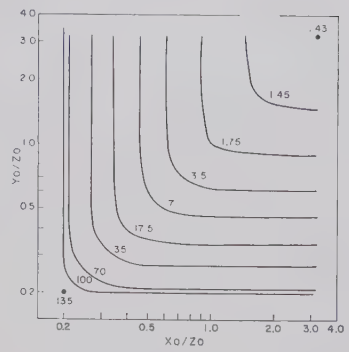


Fig. 13—Normalized standard deviation $\sigma z_0^3/M$ for magnet array of magnets as function of relative spacings (x_0/z_0, y_0/z_0). The magnets are assumed to be statistically independent and to have "ones" and "zeros" randomly.

TABLE II

Failure Probability as a Function of Normalized Field Parameter h*

h	$P(h)$
0	0.5
1	0.158
2	2.27×10^{-2}
3	1.36×10^{-3}
4	3.14×10^{-5}
5	2.78×10^{-7}
6	1.01×10^{-9}
7	1.28×10^{-12}
8	0.673×10^{-15}
9	1.19×10^{-19}
10	0.774×10^{-23}

* See text.

Then the probability $P(h)$ that such a failure will occur is given in Table II.

In the example cited earlier, we found that $H^+ = 1.4$ oe and $H^- = +2.0$ oe. Suppose the critical failure field $H_f = +1$ oe.[6] The mean μ would be $\frac{1}{2}(H^+ + H^-) = -0.3$ oe. From Fig. 13, $\sigma z_0^3/M = 11$, or $\sigma = 0.31$ oe. Then $h = (1+0.3)/0.31 = 4.2$, and $P(h) \cong 3\times10^{-5}$. Since there are 26 bits/word, the word error rate would be less than 7.5×10^{-4}; *i.e.*, the number of words which have a permanent error from the stray fields is the total number of words N times the word error rate 7.5×10^{-4}. The error rate would be less than 7.5×10^{-4}, because the field distribution is probably much smaller than the Gaussian for $H_f = 1$ oe, and also because the bits at the word ends have fewer magnet neighbors than those inside the array.

In this example, the failure field was not much different than the largest possible fields, yet an appreciable failure rate is found. This is due to the relatively small x_0. Comparison of Fig. 13 with Figs. 7–10 indicates that large ratios of $H^{\pm}$ to σ are found with designs centered around the point $x_0/z_0 = y_0/z_0 = 1$. Here $H^{\pm}/\sigma \approx 11$ (μ is usually negligible), which implies that a ratio $H^{\pm}/H_f \sim 2$ would yield a satisfactorily small fail-

[6] The experimental critical failure field for the memory of Fig. 1 is actually 2 oe. This memory is therefore expected never to fail from this mechanism.

ure rate of $\sim 10^{-8}$ per bit. The optimum bit density is evidently obtained with $x_0 = y_0 = z_0$.

It must be emphasized again that, if the critical failure field H_f is greater than $H^{\pm}$ calculated from Figs. 7–10, *no* failures are anticipated from the stray-field mechanism.

The Virtual Solenoid Structure

The long-range stray fields can be reduced in several ways. The most obvious is to reduce the effective strength of the magnet. As can be seen in (4), except on the surface of the cone where $H_z = 0$, the magnitude of the field is directly proportional to the magnet strength M. However, at the same time, the minimum distance between a twistor and its magnet must be reduced for adequate inhibition of the switching. With the arrangement illustrated in Fig. 2, there is a mechanical limitation placed on the minimum distance because of the intervening solenoid thickness. A typical minimum distance might be 0.006 inch assuming a 0.002-inch copper solenoid and 0.001-inch insulation on the twistor cable and the copper solenoid. Clearly, it would be desirable to place the magnets inside the solenoid directly against the twistor without destroying the card changeability.

A structure which achieves this objective is shown schematically in Fig. 14. Here a copper strip is placed on an insulating board. The twistors and return wires are placed outside the strips and are secured to the plastic board. A conducting sheet serves as a magnet card. With the conducting sheet in place, virtual currents are induced in the sheet when the strip is pulsed. The magnetic field so generated is temporarily confined to the space between the driven strip and the conducting sheet resulting in a localized field acting on the twistors. A second set of twistors and a conducting sheet may be placed on the lower side of the plastic board.

The behavior of the virtual current may be examined through the use of an image model.[7] In Fig. 15 let the strip conductor carry a current I above an infinite conducting sheet of thickness t and resistivity ρ. Then, if I is suddenly applied, the field above the sheet is exactly as if it were due to the current I and an image current $-I$ located below the conducting sheet. The image current $-I$ recedes in time without changing in shape at a constant velocity $v = 2\rho \times 10^{-7}/t$, emu. The field H may be calculated as a function of time from the real current I and the image current $-I$.

Several conclusions may be drawn from this model. If the strip current is wide (width w) and close to the magnet card, the field H under the strip current initially will be proportional to I/w, but will decay quasi-exponentially to $I/2w$. However, for the permanent-magnet memory, the field decay during the twistor switching time is

[7] See for example, W. R. Smythe, "Static and Dynamic Electricity," McGraw-Hill Book Co., Inc., New York, N. Y., 2nd ed., p. 403; 1950.

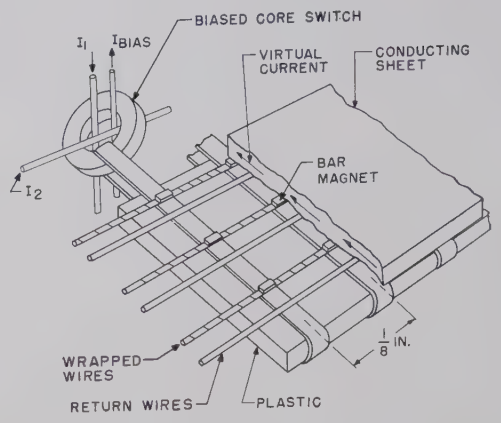


Fig. 14—Structure of a virtual-solenoid memory. The conducting sheet carries the magnets and permits the magnets to be adjacent to the twistors.

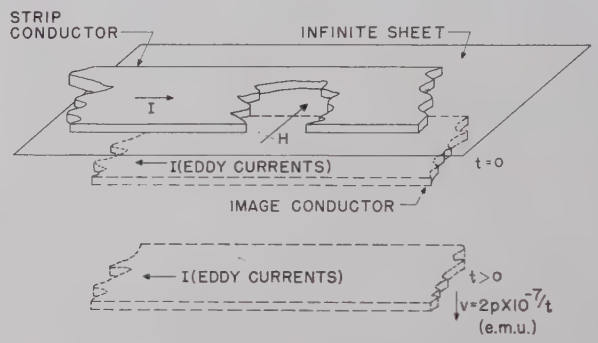


Fig. 15—Classical image model for a thin infinite conducting sheet. The field above the sheet is exactly that due to the current I and the image-eddy current. The eddy current image recedes at the constant velocity v (cm/sec), where p = sheet resistivity (emu), and t = sheet thickness.

deliberately limited. Thus a 0.005-inch copper sheet yields an image recession velocity of only 0.0085 inch per μsec.

A disadvantage of the virtual solenoid scheme is that the equivalent solenoid separation is twice that of the physical separation. This implies that the field is less uniform, and some increase in interrogation current may be necessary.

The Double-Magnet Organization

If all magnets are in position in an array, the interaction field at any location will be less than the largest possible interaction field, and can be made to approach zero field as the spacings indicated by the dashed lines on Figs. 7–10 are approached. The dashed lines indicate the spacing for equal positive and negative field.

Fig. 16 shows a scheme requiring a magnet in every position, in which a virtual solenoid is also employed. The sensing loop consists of two twistors instead of a twistor copper pair. Both are wrapped the same way and connected together at one end. To form a "one," the left-hand magnet in Fig. 16 is magnetized and its mate is demagnetized. A "zero" is formed conversely. Thus, one and only one of the magnets is always magnetized.

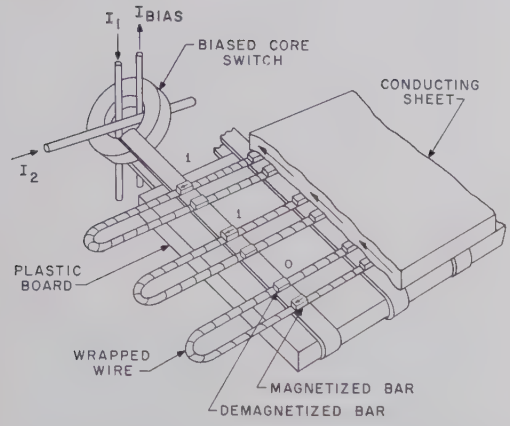


Fig. 16—Double-magnet structure. Magnetized magnet is over one or the other twistor wire to yield a "one" or "zero."

When a particular solenoid is interrogated, one or the other of the two twistor bits is free to switch, and a positive or negative pulse is observed in the output.

This organization eliminates the stray-field effects from all but the nearest-neighbor magnets. In addition the positive-one or negative-zero pulse output simplifies the sensing amplifier. The two magnet scheme requires a more complicated card-writing apparatus as well as a more difficult magnet design.

Electrical Characteristics

The permanent-magnet memory is a word-organized single-line driven memory. The input and output electrical characteristics determine the maximum size of a module of the memory and the organization of the module. In this section the largest unit of memory that is used will be called a module. It is distinguished from other memory modules by having its own associated electronics necessary to use the memory in a store. The electronic circuitry consists of the drivers, the sense amplifiers and the bias supplies. The module may be so large that, for construction reasons, it is fabricated as a number of submodules.

The biased-core switch array, considered a part of the module, reduces the driver problem from that of switching into n^2 addresses (for the single line or end-fired memory organization) to that of switching into $2n$ channels, since the cores provide coincident-current selection of the desired address. The operation here is essentially the same as in the case of the ferrite-core memory described by Rajchman.[4] Such a selection means is especially useful where the memory element contains a threshold to "pad" the operation against the relatively poor selection of this switch matrix.

The equivalent circuit seen by the current driver is shown in Fig. 17. The driver generates a voltage E_1 and a current I_1 in the biased-core switch. The input winding of the switch has a series inductance (L_1) and resistance (R_1) associated with it. The inductance is the air in-

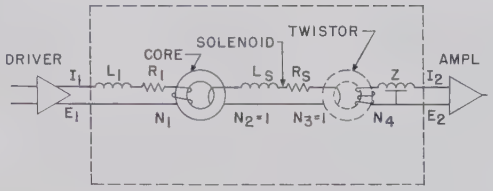


Fig. 17—Equivalent circuit of permanent-magnet memory.

ductance of the winding and the shuttling of the nonselected cores. It is approximately equal to the inductance measured by a small-signal sinewave-inductance bridge. The resistance is that of the wire. An idealized transformer is in series with L_1 and R_1, and represents the selected core. The number of turns N_1 in the primary is adjustable, whereas the secondary always has one turn. The solenoid is represented by a series inductance (L_s) and resistance (R_s) in series with the core secondary. Another transformer shown as a dashed core represents the twistor-solenoid crosspoint. The number of turns N_3 is usually 1, and N_4 is the number of turns of permalloy tape per bit length. Finally, the output is carried to the read amplifiers through a transmission line with an impedance Z.

Using the equivalent circuit of Fig. 17, which assumes that the switching resistance of the core is large with respect to the solenoid impedance and that the input ampere turns is much greater than the coercive force, the voltage E_1 at the driver is

$$E_1(p) = I_1(p)(pL_1 + R_1) + I_1(p)N_1^2(pL_s + R_s) + \frac{N_1E_0(p)N}{N_4}. \tag{8}$$

The first term represents the impedance of the biased-core switch, where p is the Laplacian operator. The next term is the voltage across the strip solenoid reflected through the core-ideal transformer (note that $N_2=1$). Finally, the voltage reflected by the twistor is represented by the last term. Here $E_0(p)$ is the output of the twistor, and so $NE_0(p)/N_4$ is the voltage reflected by N twistors (note that $N_3=1$) into the solenoid. This voltage is then reflected through the core to the driver. If the field at the twistor is large compared to the coercive force, the output $E_0(p)$ will be proportional to the net drive and (8) reduces to

$$E_1(p) = I_1(p)(pL_1+R_1) + I_1(p)(N_1)^2(pL_s+R_s) + N_1^2I_1(p)kN,$$

where k is the constant of proportionality. Since L_1 is proportional to the number of turns squared on the biased-core switch, and for a given size hole in the core, R_1 is also proportional to N_1^2, assuming that the hole in the core is filled as much as possible. One important conclusion of this simplified approach is that the drive

voltage is proportional to the input turns squared and the output to the product N_1I_1. It is possible to choose N_1 and the drive voltage and current to achieve the best electrical match for the devices used.

The signal E_2 at the amplifier is determined by the signal generated under the word solenoid, the transmission characteristics of the line consisting of the twistor-return pair, and the input impedance of the amplifier.

The signal generated under the solenoid is directly proportional to the number of turns of the permalloy tape which is switched if the demagnetizing fields can be considered small (*i.e.*, a long bit). For a fixed solenoid length and permalloy tape that is wrapped at 45° to the core wire, the output voltage is inversely proportional to the diameter of the core wire. The input signal to the amplifier will not necessarily increase as the core wire's diameter is decreased, since the attenuation of the transmission line increases as the diameter is decreased.

A typical situation is illustrated in Fig. 18. Here a twistor with a 0.003-inch diameter core wire is compared to a twistor with a 0.005-inch diameter wire. The smaller wire has $1\frac{2}{3}$ more output but the measured attenuation is 8 db/100 feet compare to 3 db/100 feet for the larger wire. There is a crossover at about 90 feet, where both twistors give the same output. For modules with longer lengths, the larger wire may be desirable.

The optimum input impedance of the readout amplifier to achieve maximum power transfer is dependent on the length of the twistor-return line. For very short memories, the impedance should be equal to the dc resistance of the line, and, for very long memories, it should be equal to the characteristic impedance of the line. If the total delay time is approximately equal to $\frac{1}{4}$ the rise time of the pulse being transmitted, the terminating impedance should lie between the dc and the characteristic impedance. This assumes that the attenuation does not decrease the signal by more than half.

The length of twistor in a given module, *i.e.*, the maximum number of words per module, is also dependent upon the output delay which can be tolerated for a given strobe interval and cycle time. The propagation time for a signal on the twistor cable is about 3.4 nsec per foot. For large modules (100 feet or more of twistor cable), the delay need not all be absorbed by a wide-strobe interval, however, since the word-selection location can anticipate the output delay and compensate for it. Although output delay is common to most magnetic memories, it should be mentioned that delay in the selection matrix must also be considered. For large arrays, this delay may be more significant.

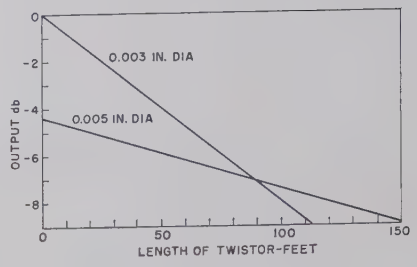


Fig. 18—Comparison of attenuation of two twistors, one made of 0.003-inch diameter copper wire and the other with 0.005-inch diameter copper wire, both wrapped with same magnetic ribbon (R_L=150 ohms.).

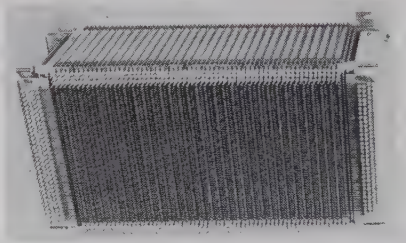


Fig. 19—Portion of a high-density submodule with the structure of Fig. 14. Metal magnet cards are inserted from the rear. This submodule contains 4096 addresses or 360,448 bits.

High Density Submodule

As one example of how some of the preceding considerations can be incorporated into a submodule design, the unit shown in Fig. 19 has been constructed. The objective of the design is to achieve a module capacity of over one million bits, random access to any address, and a cycle of 5 μsec.

The memory submodule contains 4096 addresses with 88 bits per address. Random access is provided by a 64×64 ferrite biased-core switch array. The virtual solenoid has been used in the construction, but not the two-magnet-per-bit scheme. Both the top and the bottom of the plastic mounting plate (see Fig. 14) were used, one word on each side, forming 2 words of 44 bits each per address. Such a design allowed the complete word to be on a single magnet card, with 64 words per card. It is anticipated that at least four submodules would be used to form a module of 1.4×10^6 bits capacity.

The magnet is Vicalloy I bonded on to an aluminum sheet. A slightly weaker magnet than that used for the stray-field measurements was chosen. It consists of two bars 0.035 inch long with a remanent magnetization of 5000 gauss and a coercive force of about 200 oe. Its characteristics are determined by a precipitation-hardening heat treatment. The transverse interaction is similar to that of the double-bar magnet in Fig. 4 if the memory is operated in Mode I, and the longitudinal interaction is similar to the 0.035-inch magnet in Fig. 6, but slightly wider and flatter for Mode I, as suggested by Fig. 5.

The twistor for this submodule has a 0.005-inch diameter copper-core wire wrapped with permalloy tape 0.0003 inch×0.005 inch. The return wire is 0.005-inch diameter copper wire spaced 0.020 inch from the twistor.

The cadmium ferrite cores used in the biased-core switch were arranged in a plane array 64×64, the dimensions of which were 16 inches×8½ inches. Selection currents are supplied to the multiturn windings of the access switch by transistor drivers.[2,3] The 4096 output signals of one twistor wire are shown in Fig. 20. The signal-to-noise ratio for this submodule containing 360,448 bits is comparable to that of the earlier module (Fig. 1) containing only 13,312 bits.

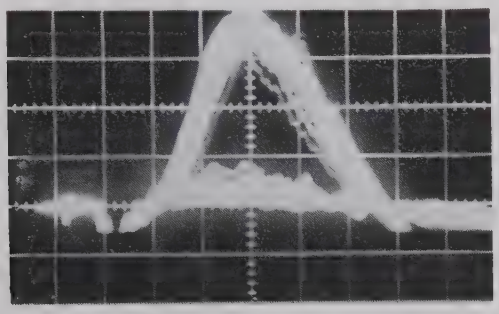


Fig. 20—Open-circuit output voltages on one wrapped wire of the 4096-addresses submodule showing 2048 zeros and 2048 ones. Each major horizontal division is 0.5 μsec., and each major vertical division is 1.0 mv.

Conclusion

The permanent-magnet-twistor memory makes available semipermanent high-capacity storage for special-purpose digital computers. Considerable flexibility is allowed in the design of such a memory. The design criteria for the permanent magnet are understood. The role of bit spacing has been studied and understood in terms of long-range interaction. The field present for a given magnet and spacing has been analyzed statistically to indicate the reliability expected from a given set of conditions. Finally, the electrical characteristics have been discussed.

Acknowledgment

The authors would like to thank D. C. Clemons and E. A. Collyer for their assistance in obtaining the data presented here, and A. J. Munn, who is responsible for the mechanical design of the submodules. The fabrication of the 4096 address submodule was completed by U. F. Gianola and J. J. Madden. Finally, we would like to thank D. H. Looney for his continued encouragement and support.

Correction

Roy G. Saltman, author of "Reducing Computing Time for Synchronous Binary Division," which appeared on pp. 169–174 of the June, 1961, issue of these Transactions, has called the following to the attention of the *Editor*.

1) Eq. (24), p. 174, should read as follows:

$$X_k = rX_{k-1} - (q_k + 1)D. \tag{24}$$

2) Eq. (25), p. 174, should read as follows:

$$q_k = m_k - 1, \qquad \operatorname{sgn} X_{k-1} = \operatorname{sgn} D$$
$$\operatorname{sgn} X_k \neq \operatorname{sgn} D. \tag{25}$$

3) On p. 174, symbols appearing as $X_{k'-1}$ should be altered to appear as X'_{k-1}, and symbols appearing as $X_{k'}$ should be altered to appear as X'_k. (The prime refers to the X and not to the k.)

The Simulation of Cognitive Processes: An Annotate Bibliography*

P. L. SIMMONS† AND R. F. SIMMONS‡

Introduction

WITH the introduction of the cybernetic point of view in the 1940's, principally by Wiener and by Shannon, a great impetus was given to interdisciplinary interest in the organic nervous system and possible analogs of it. As often has been the case with newly appreciated knowledge, the first wave of over-optimistic generalizations from the feedback principle resulted in many superficial analogies of mechanical systems to organic control systems. Too many aspects of human behavior were at one time or another described in terms of "just feedback loops." Psychoses and neuroses were described as oscillatory circuits out of control. Similarly, the organic analog was applied to the newly growing computer field. ENIAC was a "great brain." A random malfunction in computer circuitry gave rise to the thought of "neurotic" computers.

As the first wave of optimistic over-generalization receded, however, it was replaced by a steady flow of serious effort to apply the newly appreciated concepts of cybernetics, information theory and communication nets to the further understanding of organic behavior. The mathematical biophysicists, led by Rashevski, McCulloch, and Culbertson, applied cybernetic logic to the development of symbolic nerve nets for the understanding of possible methods by which a neuron system could deal with complex behaviors. Eventually, they developed their formulation to the point of feeling confident that any behavior could be symbolically described by a suitable series of nerve nets.

Mathematicians, such as Turing and von Neumann, applied their thinking to mathematical models of ultimate machines such as the Turing machine and automata. Engineers gave up the superficial analogies of computers to brains and began seriously to solve the problem of computer input-speed limitations by designing machines that could scan and translate print and other patterns into electronic analogs usable by computers. The eventual electronic speeds of inputs by scanning, as compared to the 150 cards per minute via card input, were highly attractive motivating forces causing engineers to study rather carefully the extent of information available on how the human organism recognized patterns. Although the resulting equipment resembles the human eye and nervous system no more than airplanes resemble birds, there is actually a significant transfer of some of the known principles of human pattern-recognition logic to the machines.

Computer programmers, too, soon became unenamoured with superficial analogies of malfunctions to neuroses and, instead, became aware of a more significant set of computer capabilities. The rapid development of the concepts of subroutines, macro-instructions, master programs and automatic programming gave rise to an appreciation of the importance of a program system organizing its own bookkeeping functions of storing tabular data, and even using heuristic subroutines where no algorithm was available for the transformation required. The similarity of self-organizing systems to organisms was obviously not trivial, and such functions of organisms as learning and stimulus generalization came to have the obvious implications of simpler programming procedures and the ability to accept classes of related data not previously specified in detail in the programs.

From an almost purely theoretical view, British scientists began in the forties to build experimental models of such self-organizing machines as Grey-Walter's genus of "Machina Speculatrix" and Uttley and Pask's conditional-probability learning machines. Although theoretical in orientation, their logic led rather rapidly to SAKI, a teaching machine for card-punch operators, which not only selects exercises on the basis of the operator's level of skill, but also learns from its own experience which exercises are appropriate for which skill levels. The pattern-recognition machines produced by Solartron of England also use this conditional-probability learning logic to permit pattern generalization.

In short, it appears that the enthusiastic burst of interdisciplinary applications of cybernetics to all types of control systems, including the organic, on the one hand has given way to a steadily increasing growth of organized study devoted to developing machine capabilities for self-organizing and cognitive behaviors previously thought to be exclusively the domain of organisms; on the other hand it has focused a great deal of effort from behavioral and life scientists on the analysis and synthesis of organic behaviors in terms of machines

* Received by the PGEC, December 1, 1960; revised manuscript received, February 20, 1961.

† Technical Library Services, System Development Corp., Santa Monica, Calif.

‡ Center for Research in Systems Sciences, System Development Corp., Santa Monica, Calif.

(whether mathematical, electronic analog, or digital simulation), which can simulate or synthesize them.

The resulting literature of simulated cognitive processes is large and growing rapidly. Fig. 1 shows the growth rate of a representative sample of publications in this area. Of the 330 articles and books diagrammed, 122 were published in 1959. Four years ago, in 1956, only about one-third of that number were found which met our criteria for inclusion. Prior to 1949, only a trickle of two or three articles a year was found and, of course, these were primarily theoretical in nature. In passing, it should be noted that various learning analogs were built in the 1920's and one logic machine in the nineteenth century.

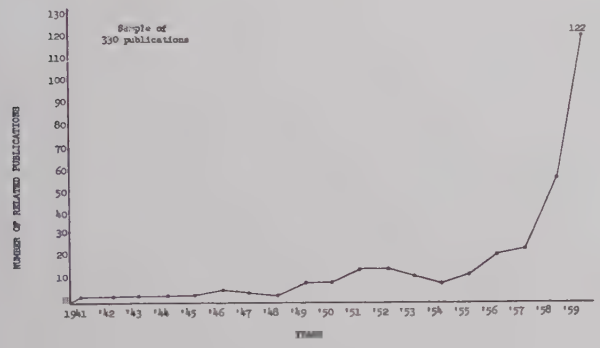


Fig. 1—The growth of publications in simulation of mental processes, from 1941–1959.

In contrast, today's literature is made up, in almost equal parts, of mathematical or logical formulations of considerable sophistication, examples of moderately successful cognitive programs, and engineers' descriptions of pattern recognition and machines with self-organizing capabilities, including electronic and chemical analogs of neurons. The trend has been toward the formulation of more complete systems, toward checker- and chess-playing programs, toward general problem-solving systems, and hardware oriented toward a definite purpose. Such trends are in marked contrast to the original *Machina Speculatrix* type of design occasionally seen in the forties as a scientific curiosity.

Tomorrow's literature, it can be surmised, will probably follow the qualitative trend toward more complex systems. If the quantity of work follows the trend of publication increases noted over the last two years, the rate of progress in this relatively new area of simulated cognitive systems may be very rapid. Since it is a strongly interdisciplinary field with a balance of theoretical, behavioral, and engineering interests and, since it is centered primarily around the use of the large-scale computer as a tool, it would not be too surprising if future developments in the field uncover many currently unsuspected applications of basic research in the simulation of cognitive functions.

Content of Bibliography

This bibliography on the simulation of cognitive functions can hardly be considered a complete survey of the field. Engineering work primarily concerned with self-organizing systems which control various complex machines is not included. Most of the symbolic model-making of psychologists attempting to symbolize various of the cognitive functions has been ignored. However, several recent efforts, such as those of Hebb and Estes, Bush and Mosteller, and some of the Millers' work, have been included.

Although the work concerned with language data retrieval and language translation fits the category of literature pertinent to the area of simulating cognitive systems, it also was not included. Several adequate summaries of this field are available. We include references in this area only to such secondary sources. (See [214] and [244].) Finally, our cutoff date for this bibliography was April, 1960. A 1961 supplement is in preparation.

The bibliographic citations are arranged by author. Annotations are included for all the primary sources which the authors have read. In the case of certain secondary sources, *e.g.*, a popular article on pattern-recognition machines, usually no annotation is given. In the case of foreign language publications, no annotation is included, unless a translated abstract is available.

Each entry in the bibliography is numbered, and a subject index to these numbered entries follows the author listing. The categories are as follows:

1) *Theoretical formulations and discussions (verbal)* (discussion of thinking or learning by machines; operational definitions or suggestions for consciousness, values, thinking, learning, etc., and some practical applications).
2) *Mathematical models, automata and probabilistics* (also, mathematical learning models).
3) *Formal nerve nets.*
4) *Neurophysiology* (particularly as applied to the development of machines).
5) *Simulated neurons and organisms* (the Perceptron, Grey-Walter's *Machina Speculatrix*, etc.).
6) *Pattern recognition* (particularly of written characters and speech).
7) *Games, problem solving, and heuristics* (chess, checkers, geometry, symbolic logic, etc.).
8) *Learning systems* (programs or machines which modify their behavior as a function of experience).
9) *Language processing* (mechanical translation and information retrieval reviews and studies of language).
10) *Miscellaneous* (general cybernetics and secondary derivative articles).

Bibliography

[1] Anonymous, "Brain simulator for perception research," *Brit. Commun. and Electronics*, vol. 5, p. 985; December, 1958.
Research on the learning mechanism of an octopus was the basis for a machine resembling the nervous system of an animal. To simulate learning processes, a small computer of 30 units was built which was able to utter a sound of recognition in response to previous learning.

[2] Anonymous, "Computers can play chess (and sometimes win)," *Data Processing Digest*, vol. 3, p. 16; October, 1957.
Summarizes the experiments in chess playing performed at Los Alamos Scientific Laboratory.

[3] Anonymous, "Learning machines," *Wireless World*, vol. 65, pp. 8–9; January, 1959.
Discusses a paper presented by A. M. Andrews on "Learning machines," *Proc. Symp. on the Mechanization of Thought Processes*, Natl. Physical Lab. No. 10, Teddington, Eng.; 1958.

[4] Anonymous, "Les machines a calculer et la pensée humaine (Calculating machines and human thought)," *Rev. Gén. Scientifique*, vol. 58, pp. 5–7; 1951. (In French.)

[5] Anonymous, "A machine that learns from experience," *Data Processing* (British), vol. 1, pp. 106–112; April–June, 1959.
Solartron Electronic Group Ltd., of Thames Ditton, Surrey, England, is producing its automatic keyboard instructor (SAKI) for rapid training of keypunch operators. The device is designed to instruct the operator in the use of the keyboard.

[6] Anonymous, "Machines are smarter than I am," *Data Processing Digest*, vol. 5, pp. 18–20; October, 1959.
Discusses the application of cognitive machines to business and industry.

[7] Anonymous, "Machines as models in biological experiments," *Nature* (London), vol. 181, p. 1187; April, 1958.
Reports paper by Dr. W. Grey-Walter delivered May, 1957, before the Laboratory Animals Bureau. Shows how laboratory-prepared animal is one type of machine. Gives principles to observe in making a machine: 1) system contains more than variables, 2) model contains no redundant parts, and 3) machine exhibits planned behavior.

[8] Anonymous, "The mechanization of thought processes," *Computer Bull.*, vol. 2, pp. 92–93; April–May, 1959.
The proceedings of a recent British Computer Society Symposium on Mechanization of Thought Processes are reviewed. Information selection, medical diagnosis, character and speech recognition and artificial intelligence are among the topics discussed.

[9] Anonymous, "Memory elements in computers," *J. Am. Med. Assoc.*, vol. 171, p. 1857; November, 1959.
Reports a paper on the concept of plastic neurons presented by D. G. Willis to the Internatl. Conf. on Information Processing, Paris, France; June, 1959. See also [486].

[10] Anonymous, "New machine 'reads' handwriting," *Process Control and Automation*, vol. 7, pp. 26–27; January, 1960.
Describes the Bell Telephone Labs. experimental handwriting reader. The writer uses a wired stylus which makes contact with 15 horizontal metal strips alternatively sandwiched between insulating strips in the writing surface. Word reading is performed by a logic circuit which analyzes the sequences and number of connections made.

[11] Anonymous, "Parlez-vous prolan? It's computer 'talk'," *Oil and Gas J.*, vol. 58, p. 51; February, 1960.
Gives a brief account of Prolan (Process Language), a language oriented solely to petroleum refining processes.

[12] Anonymous, "Perceptron, a brain analogue machine," *Brit. Commun. and Electronics*, vol. 5, p. 696; September, 1958.

[13] Anonymous, "The search for patterns and recognition," *Automatic Data Processing*, vol. 1, pp. 31–34; October, 1959.
A group at Manchester University, England, has been working on the problem of computer pattern recognition, specifically, recognition of letters in whatever position or style they may be presented. See also [433].

[14] Anonymous, "Wesen und grenzen des automaten (Nature and limits of automata)," *Mathematik-Technik-Wirtschaft*, vol. 6, pp. 9–16; January, 1959. (In German.)
Electronic computers are organized on two levels, number words and order words. The latter level also includes a conditional order which is subject to a contingency. A diagram for and automation imitating the data-processing mechanism of the human mind is given, and the principle of an order to the automation is explained by ordering multiplication.

Ackoff, R. L., See [88].

[15] Allanson, F. J., "The reliability of neurons," *First Internatl. Congress on Cybernetics*, Namur, Belgium, pp. 687–694; 1956.
Examines the reliability of neurons. If they are not reliable, then how can the effect be minimized?

[16] Andrew, A. M., "Conditional probability computer," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 2, pp. 945–946; 1959.
Conditional probability computer simulates animal learning and embodies principles which are important in the development of adaptive control for industrial applications.

[17] ——, "Learning machines," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 1, pp. 475–505; 1959.
Learning machines are discussed with reference to industrial processes and process controls.

[18] Angyan, A. J., "Machina reproducatrix, an analogue model to demonstrate some aspects of neural adaptation," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 2, pp. 933–943; 1959.
Machina reproducatrix is a model which attempts to demonstrate some (not all) of the recently formulated concepts of nervous adaptation. It is based on Dr. W. Grey-Walter's *Machina Speculatrix*.

[19] Anisimov, S. F., "Chelovek i mashina; filosfskie problemy kibernetiki (Man and machine; philosophical problems of cybernetics)," Moskva, 1959, 53 pp. (In Russian.)

[20] ——, "Man and Machine (Philosophical Problems of Cybernetics)," Joint Publications Research Service, Washington, D. C., Rept. No. JPRS 1990-N; November, 1959, 53 pp. (In English.)
A general type of discussion on the relationship of man and machines with regards to the cybernetic approach. Outlines developments in biology, sociology and artificial intelligence with the use of the computer BESM and others similar to it.

[21] —— and A. Vislobokov, "Some philosophical problems of cybernetics," *Kommunist*, vol. 36, pp. 108–118; January, 1960. (In Russian.)

Arbuckle, T., See [52].

[22] Arnold R. F., "A compiler capable of learning," *Proc. WJCC*, San Francisco, Calif., March 3–5,1959; pp. 137–143.
Concerns adaptation of learning models to the compiler and its application to learning-type compilers. The difficulty with this type of computer lies in the undeveloped capacities of machine design and programming.

[23] Ashby, W. R., "Can a mechanical chess player outplay its designer?" *Brit. J. Phil. Sci.*, vol. 3, pp. 44–47; May, 1952.

[24] ——, "Design for a Brain," John Wiley and Sons, Inc., New York, N. Y.; 1954.
Author proposes ideal adaptive system. Implies a learning process in the sense that biological controls adapt to their environment by compensating for any change that tends to make them unworkable. Homeostat is described as a device capable of random adjustment of its own parameters whenever system variable measurements exceed certain levels.

[25] ——, "Design for a brain," *Electronic Engrg.*, vol. 20, pp. 379–383; December, 1948.
An article on the Homeostat and its adaptiveness.

[26] ——, "Design for an intelligence amplifier," in "Automata Studies," C. E. Shannon and J. McCarthy Eds., Princeton University Press, Princeton, N. J., pp. 215–234; 1956.
Examines the question as to whether or not man can design a machine whose intelligence will transcend his own. Naturally he believes that this is feasible, and in order to be achieved, an intelligence amplifier should be designed to select appropriate statements from a random generation of information.

[27] ——, "Mechanical chess player," *Trans. Ninth Conf. on Cybernetics*, New York, N. Y., pp. 151–154; 1953.

Discusses question as to whether or not a mechanical chess player can outplay its designer. Problem: How is machine to develop better criteria of judgment than the designer himself can produce?

[28] ——, "The mechanism of habituation," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 1, pp. 95–113; 1959.

Discusses habituation as a phenomenon not restricted to organisms, but characteristic of any material which belongs to a class which does not totally destroy the pattern of operations applied to it. Demonstrates habituation in a set of operations applied to a random number table.

[29] ——, "The nervous systems as a physical machine: with special reference to the origin of adaptive behavior," *Mind*, vol. 56, pp. 44–59; January, 1947.

An attempt is made to solve the problem of how a physical system can show adaptive behavior. A special mathematical technique for the exact description of the properties of a machine is constructed, and a number of new theorems on equilibrium and organization in generalized dynamic systems are stated. Of special importance in the theory are the events which occur when a machine breaks down.

[30] ——, "The physical origin of adaptation by trial and error," *J. Gen. Psychol.*, vol. 32, p. 13; 1945.

Attempts to show that if any environment begins to break down any machine, the machine must exhibit essential features of trial and error by adapting and reaching a state of equilibrium.

[31] ——, "Requisite variety and its implications for the control of complex systems," *Cybernetica*, vol. 1, pp. 83–99; 1958.

Recent work by Ashby shows that fundamental processes of regulation in biology have importance of a quantitative law of requisite variety; this is related to Shannon's theorem on noise. Ashby attempts to indicate the implications for regulation in the cybernetic sense of the two theorems above.

[32] ——, "A simple computer for demonstrating behavior," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 2, pp. 947–949; 1959.

[33] ——, W. Grey-Walter, M. A. Brazier and W. R. Brain, "Perspectives Cybernetiques en Psychophysiologie," in "Travaux de Psychologie et Sciences Humaines," Presses Universitaries de France, Paris; 1951. (In French.)

[34] Atkinson, R. C., "A Markov model for discrimination learning," *Psychometrika*, vol. 23, pp. 309–322; December, 1958.

Presents a quantitative theory of discrimination learning using the concept of an observing response. In other respects, it is analogous to the models of Burke and Estes or Bush and Mosteller. The implications of the theory are derived in detail for the case where two kinds of stimuli are possible. Results of experiments with regular and partial reinforcements are compared to predictions from the model.

[35] Atkinson, R. C., "A stochastic model for rote serial learning," *Psychometrika*, vol. 22, pp. 87–95; March, 1957.

A detailed model for the acquisition of responses in an anticipatory rote serial learning situation is developed. Some predictions are derived and checked against experimental data.

[36] Bachrach, A. J., F. W. Banghart and E. G. Pattishall, "Human behavior: capacity, communications and control," *Neuropsychiatry*, vol. 4, pp. 59–79; 1956–1957.

Imbedded in the feedback circuitry of a closed-loop system are the three variables of capacity, communications and control, which are involved in an operational model of human behavior. Breakdown in efficiency can be attributed to functional interference with the feedback system as in faulty learning, or to organic interference, as in biochemical disturbance or organic destruction.

[37] Backus, J. W., "Automatic programming: properties and performance of FORTRAN systems I and II," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol 1, pp. 233–248; 1959.

The final portion of the paper discusses relationship of automatic programming and mechanization of thought processes.

[38] Baernstein, H. D. and C. L. Hull, "A mechanical model of the conditioned reflex," *J. Gen. Psychol.*, vol. 5, pp. 99–106; 1931.

A mechanical model of the conditioned reflex is presented. No claim is made that this is a duplication of the organic processes. If this can be done, however, the ultimate might be a wide range of mechanisms which manifest adaptive behavior.

Balashek, S., See [107] and [117].

[39] Balescu, I., "Future possibilities and limitations in reproducing brain mechanism by electronic machines," *Rev. Sci. Sociales (J. Social Sci.)*, Academia Republicii Populaire Romine (Bucharest, Roumania), vol. 1, pp. 45–65; 1956.

Banghart, F. W., See [36].

[40] Barlow, H. B., "Sensory mechanisms, the reduction of redundancy and intelligence," *Proc. Symp. on the Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 2, pp. 537–559; 1959.

Suggests vast amount of sensory information handled by successive codings which reduce redundancy. Includes design of a reversible binary code for this purpose. Discusses "heuristics" of perception.

[41] Barrett, E. and G. Post, "Introduction to some principles of applied cybernetics," *J. Psychol.*, vol. 30, pp. 3–10; March, 1950.

An amazing similarity exists between purposeful, teleological machines and the human being. The basic differences exist only with regard to the determination of goals. Machines are built and their goals set by men. In the human brain, goals are determined by motivations and achievement of goals induced by an effective tone or feeling tones that are the driving forces for behavior.

Basore, B. L., See [490].

[42] Beer, S., "Cybernetics and Management," John Wiley and Sons, Inc., New York, N. Y.; 1959. 214 pp.

Note especially the section on "Biophysical Theory of Cybernetics," with two chapters on machines that learn and teach, and intelligent machines for adaptation purposes. An excellent, though basic, book on cybernetics, with emphasis on management.

[43] Belenesky, I. N., "Cybernetics and some problems of physiology and psychology," *Voprosy Filosofii*, vol. 11, pp. 153–166; 1957. (In Russian.)

[44] Bellman, R., "On the Concepts of a Problem and Problem-Solving: Memorandum of Project Michigan," The University of Michigan, Willow Run Labs., Ann Arbor, Rept. No. 2144–352-R; March, 1959. 25 pp.

[45] —— and R. Kalaba, "On adaptive control processes," 1959 IRE National Convention Record, pt. 4, pp. 3–11.

The area considered here is automatic control or the design of automatic controls that learn to improve their performance based on experience or which can adapt themselves to circumstances as they find them. Uses mathematical concept of dynamic programming as the main tool.

[46] ——, ——, "On communication processes involving learning and random duration," 1958 IRE National Convention Record, pt. 4, pp. 16–21.

Special problems in prediction theory are given. Aims to show how quite general processes can be treated in unitary fashion by the functional-equation technique of dynamic programming.

[47] Belskaya, I. K., "Electronic computer study of English syntax patterns," *NBS Tech. News Bull.*, vol. 41, pp. 84–86; June, 1957.

A study of the frequency of usage of syntax patterns.

Belsky, M. A., See [52].

[48] Bemer, R. W., "A checklist of intelligence for programming systems," *Commun. Assoc. Comp. Mach.*, vol. 2, pp. 8–13; March, 1959.

Attempts to begin a systematic classification of the various devices, such as processors, supervisors and languages, that help a computer take over the decision-making functions of human operations.

[49] Bennett, G. K. and L. B. Ward, "A model of the synthesis of conditioned reflexes," *Am. J. Psychol.*, vol. 45, pp. 339–342; 1933.

A learning mechanism was demonstrated which closely resembles certain forms of animal behavior which is usually described as trial-and-error learning. This device has only two responses possible to any situation.

[50] Berkeley, E. C., "Symbolic Logic and Intelligent Machines," Reinhold Publishing Corp., New York, N. Y.; 1959. 203 pp.

The main purpose of this book is to make clear the nature of symbolic logic and the properties of intelligent machines, and to show the interrelationship of the two subjects. It is also the intention of this book to show the application of symbolic logic in the design of machines that behave intelligently.

[51] Bernstein, A. and M. de V. Roberts, "Computers vs chess player," *Sci. Am.*, vol. 198, p. 96; June, 1958.

Reports a chess-playing computer program which was written for the IBM 704 computer. This is to be played by the computer against a human opponent.

[52] ——, ——, T. Arbuckle, and M. A. Belsky, "A chess playing program for the IBM 704," *Proc. WJCC*, San Francisco, Calif., March 3–5, 1959; pp. 157–159.

Describes a program which enables a person to play a game of chess with a computer. The machine is capable of playing a complete game of chess, including such moves as castling, promoting and capturing *en passant*, and plays a passable amateur game during the opening and middle game but is weak in finishing. Though the machine always makes the same move in any given situation, these are so numerous that a game is not predictable.

[53] Beth, E. W., "On machines which prove theorems," *Simon Stevin Wisen Naturkundig Tijdschrift* (Groningen-Djakarta), vol. 32, pp. 49–60; 1958. (In Dutch.)

This is a discussion of the problem of having theorems proved by machines, the use of a decision method being restricted either arbitrarily, as in the case of the calculus of propositions, or because no such method exists, as in the lower predicate calculus. The author stresses the possible use of a semantic tableaux in this connection.

[54] Beurle, R. L., "Storage and manipulation of information in brain," *J. IEE*, vol. 5, pp. 75–82; February, 1959.

Summary review of available empirical information on interaction of cells in central nervous system and how this system reacts to incoming stimuli; interaction between cells; trial-and-error learning principle; conditioned-response principle, regenerative-memory principle.

Biddulph, R., See [108].

[55] Blachman, N. M., "NPL Symposium on the Mechanization of Thought Processes," U. S. ONR, London, Eng., Tech. Rept. ONRL-C-59; January, 1959, 10 pp.

[56] Boring, E. C., "Mind and mechanisms," *Am. J. Psychol.*, vol. 59, pp. 173–192; April, 1956.

Each computer program, to the extent that it is successful, is *de facto* a model for human cognitive process in that particular situation.

[57] Bouricius, W. G. and J. M. Keller, "Simulation of human problem solving," *Proc. WJCC*, San Francisco, Calif., March 3–5, 1959; pp. 116–119.

Describes a few pitfalls faced by anyone who tries to program machines to think or to simulate human problem solving.

[58] Bradner, H., "A new mechanical learner," *J. Gen. Psychol.*, vol. 17, pp. 414–419; 1937.

A mechanical learner which is capable of making an error immediately after it has made a correct choice by the incorporation of an inhibiting stimulus which disturbs the existing habit patterns. The working principles are those of Hull's goal-gradient hypothesis and the effect of chance external stimuli. Learning can be greatly facilitated if the machine is rewarded for correct choice.

Brain, W. R., See [33].

[59] Braitenberg, V., E. R. Caianiello, F. Lauria, and N. Onesto, "System of coupled oscillators as functional models of neuronal assemblies," *Nuovo Cimento*, vol. 11, pp. 278–282, January 16, 1959. (In Spanish.)

[60] Braynes, S. N. and A. V. Napalkov, "The brain and cybernetics," *Nauka i zhion'* (Science and Life), vol. 6, pp. 17–21; June, 1959. (In Russian.)

[61] ——, ——, "Certain Problems in the Theory of Self-Organizing Systems," Joint Publications Research Service, Washington, D. C., Rept. No. JPRS: 2177-N; January 25, 1960. 13 pp. (In English from Russian.)

A successful solution of the problem of self-organizing systems is intimately connected with our approach to certain basic problems in materialistic philosophy. In the view of the authors, the task of investigation consists in the creation of models of two systems which differ in their natures, one of which must possess a system of fixed laws, and the other an algorithm which ensures the demonstration of these laws and the evolution on this basis of new work programs.

[62] ——, ——, and Yu. A. Shrieder, "Analysis of the working principles of some self-adjusting systems in engineering and biology," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France, June 15–20, 1959, pp. 290–303; 1960.

In comparing the algorithms with the principles of cybernetic machines, there emerge certain possibilities for the construction of complex technical control systems, where the full information about the controlled objects is lacking.

Brazier, M. A., See [33].

Breido, M. G., See [219].

[63] Breydo, M. and M. Gurfinkel', "Thought runs the machine," *Tekhnika Moledezhi*, vol. 26, pp. 3–4; 1958. (In Russian.)

Principles of operation are described for a model of a human hand that can be manipulated by thought. The model was displayed at the Soviet Pavilion during the International Exhibition at Brussels. The model operates by the interception of the bio-currents that occur in muscles in response to commands issued by the brain.

[64] Broida, D., "Recent work on reading machine for data processing," *Automation Progress*, vol. 4, pp. 183–185, 224–225; May–June, 1958.

[65] Brooker, R. A., "Some technical features of the Manchester Mercury autocode programme," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 1, pp. 203–224; 1959.

Discusses the logical features of the autocode.

[66] Brown, J., "Information, redundancy and decay of the memory trace," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., vol. 2, Her Majesty's Stationery Office, London, pp. 729–745; 1959.

Theory proposed is that the memory trace is subject to decay, but the effect of decay depends on the amount of initial redundancy in the trace, on the coding of information in the trace, and on the information from other traces. This seems to provide a possible explanation of human remembering and forgetting in application to machines.

[67] Bullock, T. H., "Neuron doctrine and electrophysiology," *Science*, vol. 129, pp. 997–1002; April 17, 1959.

Discusses various new concepts that have arisen about how our nerve cells act alone and together. Points out how "all-or-none" concept is of limited utility.

[68] Burke, C. J. and W. K. Estes, "A component model for stimulus variables in discrimination learning," *Psychometrika*, vol. 22, pp. 133–145; June, 1957.

From a set-theoretical stimulus model developed by Estes and Burke, together with a descriptive theory of reinforcement, it is possible to derive a model for certain aspects of discrimination learning.

[69] Burks, A. W., "The logic of fixed and growing automata," *Proc. Internatl. Symp. on Theory of Switching*, Harvard University, Computation Lab., Harvard University Press, Cambridge, Mass. pt., I, pp. 147–188; 1959.

Author is concerned with discrete synchronous and deterministic computers. Defines and discusses fixed automata in Sections two through five, and in Section six, growing automata.

[70] ——, "The Logic of Fixed and Growing Automata," The University of Michigan, Willow Run Labs., Ann Arbor, Rept. No. 2144-231-T; February, 1958, 44 pp.

[71] ——, and H. Wang, "The logic of automata," *J. Assoc. Comp. Mach.*, vol. 4, pp. 193–218, April; pp. 279–297; July, 1957.

Mainly concerns a special class of automata, digital computers and nerve nets. Presents an interesting concept of indefinitely growing automata, which include the Turing machine.

[72] Burros, R. H., "Some criticisms of 'a mathematical model for simple learning'," *Psychol. Rev.*, vol. 59, pp. 23–236; May, 1952.

Points out a fallacy in the reasoning of Bush and Mosteller which has no effect upon the further deductions of their theory, but which might invalidate future work.

[73] Bush, R. R., "Some problems in stochastic learning models with three or more responses," in "Mathematical Model of Human Behavior: Proceedings of a Symposium," J. W. Dunlap, Ed., Dunlap and Associates, Inc., Stamford, Conn., pp. 22–24; 1955.

[74] ——, and F. Mosteller, "A mathematical model for simple learning," *Psychol. Rev.*, vol. 58, pp. 313–323; September, 1951.

Discusses a mathematical model to describe simple learning situations, with special attention to acquisition and extinction of behavior habits in the straight runway and the Skinner box.

[75] ——, ——, "A model for stimulus generalization and discrimination," *Psychol. Rev.*, vol. 58, pp. 413–423; November, 1951.

Describes a model based on elementary concepts of mathematical set theory. This model provides one possible framework for analyzing problems in stimulus generalization and discrimination learning.

[76] ——, ——, "A stochastic model with applications to learning," *Ann. Math. Statistics*, vol. 24, pp. 559–585; September, 1953.

A stochastic model designed for analyzing data with changing probabilities is presented. On each of a series of trials, one or two

alternatives occur and the probabilities of occurrence are changed from time to time by events.

[77] ——, ——, "Stochastic Model for Learning," John Wiley and Sons, Inc., New York, N. Y.; 1955. 365 pp.
Gives: Basic model; stimulus sampling and conditioning; sequences of events; distribution of response probabilities; the equal alpha condition; approximate methods; operators with limits zero and unit; commuting operators.

[78] Butz, J. S., Jr., "Electronic device simulates processes of human brain," *Aviation Week*, vol. 69, pp. 60–71; July 7, 1958.
Machine called a Perceptron, developed by Dr. Frank Rosenblatt, appears to learn the proper response to the stimulus.

Cahn, L., See [216].

Caianiello, E. R., See [59].

[79] Campaigne, H. "Some experiments in machine learning," *Proc. WJCC*, San Francisco, Calif., March 3–5, 1959; pp. 173–175.
Since the development of automatic sequence computers, it has been possible for the computer to modify its own instructions. The objective of this paper is to abbreviate the process.

[80] Campbell, D. T., "Adaptive behavior from random response," *Behavioral Sci.*, vol. 1, pp. 105–110; April, 1956.
Discusses Ashby's Homeostat with reference to various learning theories and Grey's tortoise. Feels Homeostat is more like a plant than an animal. Describes the Homeostat and discusses its functions.

[81] Carr, J. W., III, "Language, logic, learning and computers," *Computers and Automation*, vol. 7, pp. 21–22, 25–26; April, 1958.
Discusses the venturing of computers into the area of learning theory. The computer allows the human being to synthesize, not merely to analyze, the learning process.

[82] Ceccato, S. "La machine qui pense et qui parle," *Congrés Internatl. de Cybernetique* (*1st Internatl. Congress on Cybernetics*) Assoc. Internatl. de Cybernetique, Namur, Belgium, pp. 288–299; 1958. (In French.)

Chandy, J., See [405].

[83] Chauchard, P., "Artificial thinking," *Presse Medicale* (Paris), vol. 61, p. 1263; October 3, 1953. (In French.)

[84] ——, "Cybernétique et physiologie de la conscience," *Cybernetica*, vol. 2, pp. 108–113; 1958. (In French.)
Explores in detail the psychology of the human brain with the techniques of Pavlov and de Lapicque, also the oscillographic description of the nerves and reflexes by Sherrington.

[85] ——, "Psycholophysiologic comparison of thinking machines," *Presse Medicale* (Paris), vol. 58, pp. 805–806; July 8, 1960. (In French.)

[86] Chlenov, L. G., "Cybernetics in Neurology," Office of Technical Services, Washington, D. C., Rept. No. 59-13793, JPRS: L-907-N; August 17, 1959. 14 pp. (Translation of *Zhurnal Nevropatologii i Psikhiatrii*, vol. 58, pp. 1259–1264; 1958.)

[87] Chow, C. K., "Optimum character recognition system using decision function," 1957 IRE WESCON Convention Record, pt. 4, pp. 121–129.
The character recognition problem, usually resulting from character being corrupted by printing deterioration and/or inherent noise of the devices, is considered from the point of decision theory.

[88] Churchman, C. W. and R. L. Ackoff, "Purposive behavior and cybernetics," *Social Forces*, vol. 29, pp. 32–39; 1950.
The concepts and criteria of purposive behavior in cybernetics are analyzed in relation to viewpoints in psychological theory and research, and certain modifications are suggested.

[89] Clark, W. A. and B. G. Farley, "Generalization of pattern recognition in a self-organizing system," *Proc. WJCC*, Los Angeles, Calif., March 1–3, 1955; pp. 86–91.
A random neuron net self-organizing system is described. Two further experiments to determine its properties have been carried out. The first demonstrates that self-organization still takes place, even if the input patterns are subjected to considerable random variation. The second experiment indicates that after organization with the usual fixed patterns, the system classifies criterion.

——, See [127].

[90] Coborn, H. E., "The brain analogy," *Psychol. Rev.*, vol. 58, pp. 155–178; May, 1951.
The brain analogy is a theory of behavior, an instrument for psychological and physiological research, and a specification for the design and construction of mechanisms possessing intelligence.

[91] Copi, I. M., C. C. Elgot and J. B. Wright, "Realization of events by logical nets," *J. Assoc. Comp. Mach.*, vol. 5, pp. 181–196; April, 1958.
Presents a new formulation and new proofs of Kleene's work. Uses simple concepts and constructs nets out of more familiar and convenient elements. See also Kleene [217], [218].

[92] Cossa, P. "La Cybernétique. Du Cerveau Humain aux Cerveau Artificiels (Cybernetics, Human to Artificial Brains)," Masson et Cie., Paris, France; 1955. (In French.)

[93] ——, "La cybernétique ou l'art de tenir le mécanisme pour démontré (Cybernetics or the art of holding mechanisms for demonstration)," *Annee Medicale Psychologique* (*Ann. Med. Psychol.*), vol. 2, pp. 1–10; March, 1950. (In French.)
The mind is a complex electronic mechanism or robot. Cybernetics demonstrates mechanical homeostasis by reproducing the epiphenomenal and by omitting the indeterminates.

[94] Couffignal, L., "La cybernétique," *Structure et Evolution des Techniques*, vol. 8, pp. 1–7; February–April, 1957. (In French.)

[95] ——, "Les Machines á Penser (The Thinking Machine)," Les Editions de Minuit, Paris, France; 1952. (In French.)

[96] ——, "Les machines semantiques (Semantic machines)," *Proc. Ier Congres Internatl. de Cybernetique* (*1st Internatl. Congress on Cybernetics*), Assoc. Internatl. de Cybernetique, Namur, Belgium, pp. 128–138; 1958. (In French.)

[97] Cragg, B. G. and H. N. V. Temperley, "The organization of neurones: a cooperative analogy," *Electroencephalog. and Clin. Neurophysiol.*, vol. 6, pp. 85–92; February, 1954.
This paper explores the analogy between the organization of neurones and the kind of interaction among atoms which leads to cooperative processes. This analogy is found to predict some definite neurological properties which can be tested experimentally.

[98] Crider, D. B., "Cybernetics: a review of what it means and some of its implications in psychiatry," *Neuropsychiatry*, vol. 4, pp. 35–58; 1956–1957.
The general problem of communication in therapy is reviewed. Also reviews the significance of information theory in psychiatry, some relations between concepts of cybernetics and controls to problems of neurotic patients and to certain aspects of the functioning of the nervous system.

[99] Culbertson, J. T., "Even in Memoryless Robots There Is No Small Number of Central Cells Sufficient for All Input-Output Specifications," RAND Corp., Santa Monica, Calif., Paper P-316; August, 1952. 9 pp.
Extends problem of finding an upper boundary to the number of central cells required for any input-output specifications.

[100] ——, "Hypothetical Robots and the Problem of Neuroeconomy," RAND Corp. Santa Monica, Calif., Paper P-296; April, 1952. 58 pp.
A study of general methods of constructing robots. The need to find more economical methods of construction (fewer neurons) is discussed.

[101] ——, "Robots and automata: a short history. Bibliography," *Computers and Automation*, vol. 6, p. 20; April, 1957.

[102] ——, "Robots and automata: a short history," *Computers and Automation*, vol. 6, pp. 32–37; March, 1957.
A charming mythological, literary and scientific history of man's attempts to create robots across the centuries.

[103] ——, "Sense Data in Robots and Organism," RAND Corp., Santa Monica, Calif., Paper P-378; May, 1953. 41 pp.
Presents a theory showing how sense data are produced by the passage of impulses through a nervous system and applies the theory to visual sense data in flat perception.

[104] ——, "Some uneconomical robots," in "Automata Studies," C. E. Shannon and M. MacCarthy, Eds., Princeton University Press, Princeton, N. J. pp. 90–116; 1956.
Attempts to show how to construct a memoryless robot. A way of introducing probabilistic behavior without inserting unreliable elements.

[105] David, E. E., Jr., M. V. Mathews, and H. S. McDonald, "Description and results of experiments with speech using digital computer simulation," 1958 IRE WESCON Convention Record, pt. 7, pp. 3–10.

[106] ——, ——, ——, "A high-speed data translator for computer simulation of speech and television devices," *Proc. WJCC*, San Francisco, Calif., March 3–5, 1959; pp. 169–172.
Describes a data translator which, when used with a digital computer, permits simulation of speech- and television-processing devices. The output can be displayed visually or audibly for subjective evaluation.

[107] Davies, D. W., "Mechanization of thought processes," *Nature*, vol. 183, pp. 225–226; January 24, 1959.
Topics discussed at this symposium are: Heuristic methods of pattern recognition; learning; and the use of analogies or models.

[108] Davis, K. H., R. Biddulph, and S. Balashek, "Automatic recognition of spoken digits," *Proc. Second London Symp. on Applications of Communications Theory*, London, Eng., Butterworth Scientific Publications, London, Eng.; 1953.
The recognizer will automatically recognize telephone quality digits at normal rate by a single individual, with an accuracy of 97–99 per cent.

[109] Davis, M. D., "A Note on Universal Turing Machines," in "Automata Studies," C. E. Shannon and J. MacCarthy Eds., Princeton University Press, Princeton, N. J., pp. 167–175; 1956.
The universality of a Turing machine is manifested by its ability to perform only computation which could be performed by any given Turing machine. However, the coding must be simple. This explains and defines a universal Turing machine.

[110] Davis, R. C., "The domain of homeostasis," *Psychol. Rev.*, vol. 65, pp. 8–13; January, 1958.
Concludes that there is no compulsion to think of the organism as an elaborate machine for the purpose of getting itself back to the *status quo ante*, or, indeed, for any other purpose.

[111] Delpech, L., "Psychoánalyse et cybernétique (Psychoanalysis and cybernetics), *Proc. Internatl. Congress of Philosophy*, XI, vol. 7, North Holland Publishing Co., Amsterdam, The Netherlands, pp. 155–161; 1953.

De Lucia, A., See [214].

Denes, P., See [139], [140].

[112] Designers for Industry, Inc., "Mechanized Processing of Engineering and Production Information: a Universal Man-Machine Coding System," Cleveland, Ohio, Summary Rept. March, 1959.
Automation is discussed in its many forms: numerical control, pattern recognition, machine language translation, and information theory.

[113] Deutsch, J. A., "A machine with insight," *Quart. J. Exp. Psychol.*, vol. 6, pp. 6–11; February, 1954.
A machine capable of demonstrating insightful behavior and learning is described.

[114] Deutsch, K. W., "Mechanism, organism, and society: some models in natural and social science," *Phil. Sci.*, vol. 18, pp. 230–252; 1951.
An excellent general discussion of the use of models in science, with applications to society and with reference to learning. Particularly valuable for clarifying concepts of will, value, belief, and consciousness.

[115] Dimond, T. L., "Devices for reading handwritten characters," *Proc. EJCC*, Washington, D. C., December 9–13, 1957; pp. 232–237.
Develops a process for reading hand-printed numerals, provided that they are drawn in accordance with certain constraints that restrict size, proportion, and location.

[116] Dineen, G. P., "Programming pattern recognition," *Proc. WJCC*, Los Angeles, Calif., March 1–3, 1955; pp. 94–100.
The model was furnished by Selfridge. Discusses the design of basic operations that the machine uses to reduce the input phase when programming pattern recognition. Machine used was Lincoln Laboratory's Memory Test Computer.

Duda, W. L., See [368].

[117] Dudley, H. and S. Balashek, "Automatic recognition of phonetic patterns in speech," *J. Acoust. Soc. Am.*, vol. 30, pp. 721–732; 1958.
A model of a phonetic pattern recognition which is based on frequency-spectrum analysis.

[118] Dunham, B., R. Fridshal and G. L. Sward, "A non-heuristic program for proving elementary logical theorems," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France, June 15–20, 1959, pp. 282–285; 1960.
A problem solver for certain types of problems would involve strategy. To what extent would systematic procedures be found useful for machine use? Results do indicate the general efficiency of the algorithmic approach for such problems.

——, See [136].

[119] Eccles, J. C., "The Neurophysiological Basis of the Mind: The Principles of Neurophysiology," Oxford University Press, New York, N. Y.; 1953. 314 pp.
Covers the field of neurophysiology and the current methods of investigation.

Edwards, D. J., See [367].

[120] Eldredge, K., "Teaching machines how to read," *J. Stanford Res. Inst.*, vol. 1, p. 1; February, 1957.

[121] Elgot, C. C., "Decision Problems of Finite Automata Design and Related Arithmetics," The University of Michigan, Research Inst., Ann Arbor, OOR Rept. 1963:2; June, 1959. 49 pp.
Explores the mathematics of automata behavior and decision design.

——, See [63].

Ellis, T. O., See [398].

[122] Ellson, D. G., "A mechanical synthesis of trial and error learning," *J. Gen. Psychol.*, vol. 13, pp. 212–218; 1935.

[123] Estavan, D. "Pattern Recognition, Machine Learning and Automated Teaching," System Development Corp., Santa Monica, Calif. Rept. No. SP-70; May 1, 1959. 8 pp.
Describes a character-recognition program using these principles for the computer: explores machine learning, presenting several definitions of learning that are broad enough to encompass this concept; further describes an SDC-formulated computer learning program.

[124] Estes, W. K., "Models for learning theory," *Symp. on Psychology of Learning Basic to Military Training Problems*, Dept. of Defense, Res. and Dev. Board. Washington, D. C.; 1953. 195 pp.

[125] ——, "Theory of Elementary Predictive Behavior; An Exercise in the Behavioral Interpretation of a Mathematical Model," in "Mathematical Models of Human Behavior," Dunlap and Associates, Inc., Stamford, Conn., pp. 63–67; 1955.
An empirical situation of considerable interest to the learning theorist is the behavior of an individual in attempting to predict the occurrence of an uncertain event. The development of this type of behavior is of obvious practical interest, and under suitably simplified conditions, is a bridge between the simple and complex learning situations.

[126] ——, "Toward a statistical theory of learning," *Psychol. Rev.*, vol. 57, pp. 94–107; March, 1950.
Attempts to clarify some issues in current learning theory by giving a statistical interpretation to the concepts of stimulus and response and by deriving quantitative laws that govern simple behavior.

——, See [68].

[127] Farley, B. G. and W. A. Clark, "Simulation of a self-organizing system by a digital computer," IRE Trans. on Information Theory, No. P61T-4, pp. 76–84; September, 1954.

——, See [89].

[128] Fatehchand, R., "Machine recognition of spoken words," in "Advances in Computers," Franz L. Alt, Ed., Academic Press Inc., New York, N. Y., vol. 1, pp. 193–229; 1960.
Gives a summary of the development in machine recognition of spoken words.

[129] Felker, J. H., "Mechanized memory and logic: what electronics can do," *Bell Labs. Record*, vol. 34, pp. 201–206; June, 1956.

[130] Findler, N. V., "Some remarks on the game 'Dama' which can be played on a digital computer," *Computer J.*, vol. 3, pp. 40–44; April, 1960.
The popularity of intelligence-simulating programs is increasing. Several games have been programmed for computing machines, with the idea of investigating machine-learning techniques. This paper describes the strategies employed in a program for playing the game of Dama on the SILLIAC. A learning process is suggested by means of which an optimal grand strategy can be achieved.

Flory, L. E., See [498].

[131] Foulkes, J. D., "A class of machines which determine the statistical structure of a sequence of characters," 1959 IRE WESCON Convention Record, pt. 4, pp. 66–73.

The machines described were originally formulated in an attempt to model human behavior in experimental situations of the type described by Bush and Mosteller.

[132] Frankel, S., "Information-theoretic aspects of character reading," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France, June 12–20, 1959, pp. 248–251; 1960.

The development of instruments for the reading of printed typed characters presents a problem in pattern recognition with several specializations. The ideal is to have instruments which will read and recognize characters and symbols.

[133] ——, "On the design of automata and the interpretation of cerebral behavior," *Psychometrika*, vol. 20, pp. 149–162; June, 1955.

Method of computational investigation of the functioning of models of human behavior is described. Several extremely simple models have been investigated, and one is shown which has properties of learning. Concludes that stupidity may result from quick learning.

[134] ——, "Proposed Neural Network Investigation," Digital Computer Group, California Inst. Tech., Pasadena; 1959.

Fridshal, R., See [118].

[135] Friedberg, R. M., "A learning machine. Part I," *IBM J. Res. & Dev.*, vol. 2, pp. 2–13; January, 1958.

[136] ——, R. Dunham, and J. H. North, "A learning machine. Part II," *IBM J. Res. & Dev.*, vol. 3, pp. 282–287; July, 1959.

An effort is made to improve the performance of the learning machine described in Part I, and the over-all effect of various changes is considered. Comparative runs without the scoring mechanism indicate that the grading of instructions can aid the machine to learn.

[137] Friedman, G. J., "Digital simulation of an evolutionary process," *General Systems: Yearbook of the Society for General Systems Research*, vol. 4, pp. 171–184; 1959.

Many similarities of behavior have been shown to exist between living and nonliving systems. Modern feedback control systems exhibit the same goal-seeking tendencies as the nervous system of an animal. Giant electronic "brains" imitate the computational and logical activity of the human mind. The study of these areas of similarity could lead to the development of more advanced and sophisticated computer-control systems and could conceivably give insight into the human system.

[138] ——, "Selective Feedback Computers for Engineering Synthesis and Nervous System Analysis," Master's thesis, University of California, Los Angeles; 1956.

[139] Fry, D. B. and P. Denes, "An analogue of the speech recognition process," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 1, pp. 377–384; 1959.

Describes an electronic analog computer which analyzes human speech sounds, prints out phoneme symbols, and recognizes as estimated 70 per cent of sounds and 45 per cent of words. Recognition logic is based on linguistic probabilities.

[140] ——, ——, "Mechanical speech recognition," *Proc. Second Symp. on Applications of Communications Theory*, London, Eng., Butterworth Scientific Publications, London, Eng.; 1953.

Speech communication is established by means of acoustic waves passing from speaker to listener. The task of a mechanical recognizer is to differentiate between the various patterns forming its input and give an output coded in 40 units. This paper describes the early attempts and considers the failure in the light of a theory of speech recognition.

[141] Gal'perin, I. I., "On the reflector nature of controlling machines," *Voprosy Filosofii* (*Problems of Philosophy*), vol. 4, pp. 158–168; July–August, 1957. (In Russian.)

Ganzhorn, K., See [417].

[142] Gardner, M., "Logic Machines and Diagrams," McGraw-Hil Book Co., New York, N. Y., 1958.

Discusses the logic of Ramon Lull, Venn circles, Stanhope demonstrator, Jevon's logic machine, Marquand's machine and the future of problem-solving and logic machines. An excellent book written for the layman but for an intelligent one. Draws from material which is not so well known.

Garfinkel', V. S., See [219].

[143] Gavrilov, M. A., "Functions of memory and comparison in automatic machines," *Priroda* (*Nature*), vol. 45, pp. 30–39; October, 1956. (In Russian.)

[144] Gelernter, H., "A note on syntactic symmetry and the manipulation of formal systems by machine," *Information and Control*, vol. 2, pp. 80–89; April, 1959.

The computer is called on to manipulate a complex formal logistic system as a tool to implement solution of a problem. This paper is concerned with the problem of machine manipulation of logic in which the system displays a high degree of symmetry.

[145] ——, "Realization of a geometry theorem proving machines," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France, June 12–20, 1959, pp. 273–282; 1960.

[146] —— and N. Rochester, "Intelligent behavior in problem-solving machines," *IBM J. Res. & Dev.*, vol. 2, pp. 336–345; October, 1958.

Describes a machine that can solve theorems in Euclidean plane geometry. Heuristic methods and learning machines are discussed, and the concept of learning theory as an extension of theorem proving is introduced.

[147] George, F. H., "Automation, Cybernetics and Society," Philosophical Library, New York, N. Y.; 1959.

See especially part two, chapter 14, entitled: "Programming a computer to learn." Semi-technical in style and coverage, but an informative book.

[148] ——, "Behavior network systems for finite automata," *Methodos*, vol. 9, pp. 279–291; 1957.

Suggests suitable models for constructing machines which are capable of performing the acts of humans and as such are models for psychological theory.

[149] ——, "Cybernetic models and their applications," *Process Control and Automation*, vol. 6, pp. 92–97; March, 1959.

Discussion of models capable of self-correcting or learning from experience; review of research carried out in Great Britain and U. S. on learning machines.

[150] ——, "Inductive machines and the problem of learning," *Cybernetica*, vol. 2, pp. 109–126; February, 1959.

A general article on induction machines and learning machines of the recent past.

[151] ——, "Logical networks and behavior," *Bull. Math. Biophysics*, vol. 18, pp. 337–348; December, 1956.

A method is given which presents a systematic account of behavior in mathematical terms.

[152] ——, "Logical networks and probability," *Bull. Math. Biophysics*, vol. 18, pp. 187–199; September, 1957.

Logical networks are given with specific motivation which acts like a reinforcer to the network. Presents ways in which the probability enters the logical network and transforms it into a probability network.

[153] ——, "Logic and behavior," *Science News*, vol. 45, pp. 46–60; 1957.

The author asserts that the need for the prediction of individual behavior as against groups of organisms can only be met by a study of internal or physiological mechanisms. Further, theories concerning these mechanisms require such symbolic logic models as McCulloch and Pitts have proposed. One advantage of regarding organisms as logic net computers is the possibility of integrating the logical, mathematical, and biological sciences.

[154] ——, "Machines and the brain," *Science*, vol. 127, pp. 1269–1274; May 30, 1958.

Discusses: How science has used cybernetics; complex nets designed by mathematical logic, whose arrangements resemble structure of brain; application of such networks to machine; what further analogies could be drawn from the nervous system.

[155] ——, "Probabilistic machines," *Automation Progress*, vol. 3, pp. 19–31; January, 1958.

Discusses: Recent developments in the field of probabilistic computers; inductive and deductive system, possible applications; reading problems; teaching and learning machines.

[156] —— and J. H. Handlon, "Toward a general theory of behavior," *Methodos*, vol. 7, pp. 25–54; 1955.

Presents a molar account of learning and perception which is intended to provide a line of research for cybernetic work.

[157] Gerard, R. W., "Some of the problems concerning digital notions in the central nervous system," *Trans. Seventh Conf. on Cybernetics*, pp. 11–57; 1960.

Contains a discussion by McCulloch, Gerard, von Neumann, Pitts, Fremont-Smith, Wiener, Stoud, and others on digital and analog functions in the nervous system.

[158] ——, "What is memory?" *Sci. Am.*, vol. 189, pp. 118–126; September, 1953.

[159] Gill, A., "Minimum-scan pattern recognition," IRE TRANS. ON INFORMATION THEORY, vol. IT-5, pp. 52–58; June, 1959.

Speedier and simpler pattern-recognition systems can be realized when provided with a minimum scan-reading device. For the idealized case, where the input set of patterns is finite, binary and errorless, a theorem is proved which enables the designer to predict the efficiency range of the contemplated reading device.

[160] Gill, S., "Possibilities for the practical utilisation of learning processes," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 2, pp. 827–833; 1959.

Suggests ways that a computer or machine which could learn may be used. One point emphasizes that a medical doctor studies years to accomplish his education only to be able to use this for a short professional period. The machine would have an unlimited life expectancy for medical diagnosis.

[161] Gilmore, P. C., "A program for the production from axioms, of proofs for theorems derivable within the first order predicate calculus," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France, June 12–20, 1959, pp. 265–273; 1960.

Program for the IBM 704 computer which will construct proofs for theorems of what is variously called the first-order predicate calculus or logic.

[162] Glantz, H. T., "On recognition of information with digital computer," *J. Assoc. Comp. Mach.*, vol. 4, pp. 178–188; April, 1957.

Whereas recognition of information or data patterns is a simple task for humans, there exists vast discrepancy between power of discrimination exercised by digital computer and that of the human.

[163] Glanzer, M., "Coding and Use of Information in Problem-Solving," University of Maryland, College Park, Progress Rept.; February 1, 1959–January 31, 1960. 4 pp.

Problem solving is analyzed as an information-processing system. A series of four experiments on concept formation is described briefly. These experiments furnish a basis for constructing a specific model for the information processing aspects of concept formation. A second series of three experiments studying set in problem solving, and other related experiments completed within last few years are given.

[164] Glauberman, M. H., "Character recognition for business machines," *Electronics*, vol. 29, pp. 132–136; February, 1956.

Goldberg, S. H., See [367].

[165] Good, I. J., "Could a machine make probability judgments?" *Computers and Automation*, vol. 8, pp. 14–16; January, 1959.

Using as an analogy the way human chess players make probability judgments in assessing their strategy, it is suggested that a computing machine with random networks may, within the foreseeable future, be making probability judgments as effective as those of experienced humans.

[166] Gorn, S., "On the mechanical simulation of habit-forming and learning," *Information and Control*, vol. 2, pp. 226–259; September, 1959.

Describes types of simulation mechanisms that may be utilized. Uses the language of computer programming to describe flow of control and mathematical probability to effect behavior in terms of simulating programs.

[167] Graham, R. E. and J. L. Kelley, Jr., "A computer simulation chain for research on picture coding," 1958 IRE WESCON CONVENTION RECORD, pt. 4, pp. 41–46.

[168] Gray, H. J., Jr., "Information Retrieval and the Design of More Intelligent Machines," Moore School of Elec. Engrg., University of Pennsylvania, Philadelphia, Final Rept. on Task E, AD 59UR1; July, 1959. 200 pp.

The specific task is one of discovering how to make data-processing systems more intelligent by the use of list techniques developed through the study of certain logical design problems.

[169] Greanias, E. C., "Design of logic for recognition of printed characters by simulation," *IBM J. Res. & Dev.*, vol. 1, pp. 8–18; January, 1957.

Describes a system for recognizing members of a sixteen-character alphabet. The system was tested via a computer program and successfully identified samples of printed characters in the presence of noise.

[170] Green, B. F., "Non-computational uses of digital computers," *Behavioral Sci.*, vol. 4, pp. 164–167; April, 1959.

Outlines some of the novel ways in which computers have been used to further psychological research.

[171] ——, "The Use of High-Speed Digital Computers in Studies in Form Perception," Mass. Inst. Tech., Lincoln Lab., Lexington; 1957. 16 pp.

[172] Greene, P. H., "An Approach to Computers that Perceive, Learn, and Reason," University of Chicago, Ill., AFOSR Rept. TN-59-375; 1959. 6 pp. (Reprinted from *Proc. WJCC*, San Francisco; Calif., March 3–5, 1959; pp. 181–186.)

Discusses a a general conceptual outlook which has important consequences for the approach to such intelligence machines. Shows how this outlook calls for modifications or supplementation of current approaches.

[173] ——, "Networks for pattern perception," *Proc. Natl. Electronics Conf.*, vol. 14, pp. 357–369; 1959.

This paper attempts to show how a certain type of artificial network could exhibit many of the pattern-stabilizing and perceiving abilities that make human vision meaningful. The model is not intended to do the entire job of recognition; rather, its purpose is to notice and hold together good gestalten, or perceptual units, in order that a pattern recognizer incorporating the networks would have stable and meaningful units with which to operate.

[174] ——, "Problem-solving and learning machines," *Behavioral Sci.*, vol. 4, pp. 249–250; July, 1959.

[175] Gregory, R. L., "Models and the localisation of function in the central nervous system," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 2, pp. 671–681; 1959.

Explains the use of conceptual models to aid in determining what functions in the nervous structure are localized.

Grey-Walter, W., See Walter, W. Grey.

[176] Grimsdale, R. L., "Automatic pattern recognition," *Wireless World*, vol. 65, pp. 499–501; November, 1959.

Discusses the ability of the human eye to see and the brain to recognize characters and patterns. By use of a new morphological system with a digital computer, it is hoped that machine recognition can be achieved.

——, See [215].

[177] Gruenberg, E. L., "The concept of thinking," *Computers and Automation*, vol. 3, pp. 18–21; April, 1954.

By use of Dewey's concept of thinking, the process can be simulated with a computer, thereby sidestepping the metaphysical system.

[178] ——, "Reflective thinking in machines," *Computers and Automation*, vol. 3, pp. 12–19, 26, 28; February, 1954.

Using John Dewey's meaning of reflective thinking, which is a series of operations on tentative suggestions, the author believes that machines have the raw material for reflective thinking.

[179] ——, "Thinking machines and human personality," *Computers and Automation*, vol. 4, pp. 6–9; April, 1955.

Since computers have no emotions and are not likely to develop them, we need have no fear of being taken over. We may expect machines to function as humans and yet be more adaptable.

[180] Grulband, G. Th., "La Cybernétique (Cybernetics)," Presses Universitaires de France, Paris; 1954. 136 pp. (In French.)

The author presents: 1) What he calls the background of an article on cybernetics for the encyclopedia; 2) Problems of servomechanisms, nets, and circuits, denoting a separated section of the Homeostat of Ashby; 3) The theory of information ("signals and messages"); and 4) Remarks on the theory of games and some general problems of cybernetics. Has a 27-item bibliography.

[181] Gurevich, B. K., "Cybernetics and certain problems of contemporary physiology of the nervous system," *Vestnik Akademiia Nauk SSSR*, vol. 24, pp. 31–40; 1957. (In Russian.)

[182] ——, "Reasoning Automats and the Higher Brain Functions," Joint Publications Research Service, Washington, D. C., Rept. No. JPRS: L-1049-N; November, 1959. 20 pp.

In examining the problem of concept formation in automats, the author points to the current tendency for the principles of cybernetic automats and those of behavioristic response theory to approach each other. Concludes that as neurophysiology shows the nervous apparatus of higher brain functions is essentially more complicated and plastic as compared with the structure of reasoning automats.

[183] ——, "Reasoning automats and the higher brain functions," *Voprosy Psikhologii* (*Problems of Psychology*), vol. 5, pp. 3–15; July–August, 1959. (In Russian.)

Gurfinkel, M., See [63].

[184] Gyr, J. W., "An investigation into, and speculations about, the formal nature of a problem-solving process," *Behavioral Sci.*, vol. 5, pp. 39–59; January, 1960.

Problem solving is viewed as a process in which a person constructs a population of possible hypotheses about his environment. selects one or a few of these for testing, and receives information from the environment regarding the truth or falsity of this tested hypothesis. Finally, he discusses the possibility of programming a computer to solve "PSI" as humans seem to do.

[185] Hagelbarger, D. W., "SEER, a sequence extrapolating robot," IRE Trans. on Electronic Computers, vol. EC-5, pp. 1–7; March, 1956.

The success of computers in doing routine work formerly done by clerks suggests that a computer capable of adjusting to the environment is desirable. As a step in this direction, a relay machine that plays a penny-matching game with human opponents has been built. Its behavior against people and other machines is discussed.

[186] Hagensick, P. W., "Logic by machine: programming the LGP-30 to solve problems in symbolic logic," *Behavioral Sci.*, vol. 5, pp. 87–94; January, 1960.

Describes in some detail a program for enabling the LGP-30 to solve problems in symbolic logic. The problems are limited to expressions in propositional logic.

Haibt, L. H., See [368].

[187] Haldane, J. B. S., "The mechanical chess-player," *Brit. J. Phil. Sci.*, vol. 3, pp. 189–191; August, 1952.

Handlon, J. H., See [156].

Hare, P. F., See [189].

[188] Harlow, H. F., "The brain and learned behavior," *Computers and Automation*, vol. 4, pp. 6–14; October, 1951.

[189] Haroules, G. G., and P. F. Hare, "Jenny: An Improved Homeostat," AF Command and Control Dev. Div., ARDC, Bedford, Mass., Rept. No. TN-60-379; April, 1960. 29 pp.

An improved and enlarged model of Ashby's Homeostat is described, in which conventional 60-cycle, electromechanical analog-computer techniques are used instead of the original dc techniques, the number of elements available as main variables is increased from four to sixteen, and provisions are made for electrical command inputs. Applications for an experimental program in adaptive machines is given.

[190] Harmon, L. D., "Artificial neuron," *Science*, vol. 129, pp. 962–963; April, 1959.

Describes an electronic model for simulating many of the gross operational functions which are believed to hold for living nerve cells.

[191] Hartmanis, J., "The application of some basic inequalities for entropy," *Information and Control*, vol. 2, pp. 199–213; September, 1959.

In recent years, there has been a rapidly growing interest in learning machines and self-adaptive servomechanisms. An important problem is to determine what information should be processed and stored. In conclusion, the results are applied to a coding problem to determine the delay required before coding a message to achieve an optimal comparison by Shannon's coding theorem.

[192] Hay, J. C., "Mark I Perceptron Operator's Manual," Cornell Aeronautical Lab., Buffalo, N. Y., Rept. VG-1196-G-5; February 15, 1960. 59 pp.

This manual was intended primarily as a guide to the setting up and operation of the Perceptron. It does not require extensive familiarity with the theory of Perceptron systems. Points out the machines capabilities as a research instrument.

[193] —— and C. W. Wrightman. "The Mark I Perceptron," *Research Trends*, Cornell Aeronautical Lab., Buffalo, N. Y., vol. 8, pp. 1–4; Spring, 1960.

A writeup on Rosenblatt's Perceptron and the experiments which have been performed on it.

[194] Heasly, C. C., Jr., "Some communication aspects of character-sensing system," *Proc. WJCC*, San Francisco, Calif., March 3–5, 1959; pp. 176–180.

Analyzes the environment in which the character-sensing machines must operate. This involves understanding the nature, the technique, and the factors which must be employed.

[195] Hebb, D. O., "The Organization of Behavior; A Neuropsychological Theory," John Wiley and Sons, Inc., New York, N. Y.; 1959.

Presents a theory of behavior based on the physiology of the nervous system and makes a sedulous attempt to find some community of neurological and psychological conceptions. Perceptual change integration is described in terms of cell assemblies.

[196] Highleyman, W. K. and L. A. Kamentský, "A generalized scanner for pattern and character recognition studies," *Proc. WJCC*, San Francisco, Calif., March 3–5, 1959; pp. 291–294.

Discusses automatic conversion of human phraseology to language understandable by a machine and its importance to the character recognition devices.

[197] ——, ——, "Pattern recognition (perception) machine," *Behavioral Sci.*, vol. 4, p. 248; July, 1959.

Hardware-oriented article on perception technique for electronic machines.

[198] Hilton, A. M., "Logic and switching circuits," *Elec. Engrg.*, vol. 65, pp. 125–158; April, 1960.

Defines logic as the science concerned with finding generalizations and, ultimately, abstractions by a system of orderly thinking. The procedure is progressive systematization and generalization, and the aim is the discovery of abstract forms. Considers the principles of logical thought and their relationship to modern computers. Discusses: the principles of symbolic logic; classes (sets); logical reasoning; the algebra of logic; logic and computing machines for control; and logic and circuit design. Contains a bibliography of approximately 80 items.

[199] Hlushkov, V. M., "Electronic calculating machines and cybernetics," *Radiani'ka Shkolo* (*Soviet School*), vol. 38, pp. 76–83; September, 1959. (In Ukrainian.)

Holland, J. H., See [368].

[200] Holland, J., "Survey of Automata Theory," The University of Michigan, Willow Run Labs., Ann Arbor, Rept. 2900-52-R; September, 1959. 24 pp.

Automata theory draws its methods and data from several fields: logic and metamathematics, theory and use of stored-program computing machines, information theory, neurophysiology and related parts of psychology. This is an exploration of the field with some conclusions on feedback.

Hull, C. L., See [38].

[201] Illinois University Elec. Engrg., Res. Lab., "The Realization of Biological Computers," Heinz Von Foerster, Ed., Quarterly Progress Rept. 5, Urbana, Ill.; January 15–April 15, 1959. 26 pp.

Discusses the topology of neural nets and the reliability of these nets. Presents abstract models and gives specifications for a concrete model.

Imoto, K., See [467].

[202] International Business Machines Corp., "IBM Lightning Project," Progress Rept. 1, December 1, 1958–March 31, 1959, Yorktown Heights, N. Y.; April, 1959. Various pagings.

This work represents a continuation of the first phase of this program. Its purpose is to determine the feasibility of a complete computer system of exceedingly high operational speeds employing superconducting circuitry and to find practical solutions to most of the major technical problems. In this effort the material under the phase of symbolic logic is especially interesting, as the work reports on machine learning experimentation.

[203] Jacobson, H., "The informational content of mechanisms and circuits," *Information and Control*, vol. 2, pp. 285–296; September, 1959.

Presents methods of calculating the information content, *i.e.*, complexity of interacting systems of parts in mechanisms and circuits. The method has been developed primarily in order to describe self-reproducing systems.

[204] Jefferson, G., "The mind of mechanical man," *Brit. Med. J.*, vol. 1, pp. 1105–1121; 1949.

[205] Jeffrey, R. C., "Some Recent Simplifications of the Theory of Finite Automata," Mass. Inst. Tech., Res. Lab. of Electronics, Cambridge, Rept. No. TR-219; May 27, 1959. 11 pp.

Discussion of a nonprobabilistic automaton for which the present input symbol and present state do not uniquely determine the next state.

[206] John, E. R., "Contributions to the study of the problem-solving process," *Psychol. Monographs*, vol. 71, no. 18; 1957. 39 pp. (Whole number 447.)

"Problem-solving and information apparatus" (PSI) was used. Utilizing this electro mechanical Boolean computer, the investigator studies problem-solving behavior of 59 University of Chicago students. He reports on significant differences in performance related to

the disciplines from where the factors are obtained, the stage of training and education they have, etc.

[207] —— and J. G. Miller, "The acquisition and application of information in the problem-solving process: an electronically operated logical test," *Behavioral Sci.*, vol. 2, pp. 291–300; October, 1957.

With the advent of electronics, thinking machines, and information theory has come the possibility that human problem solving can now be studied in a more objective way.

Johnson, D. L., See [326].

[208] Jonassen, H. B., "European Scientific Notes, 13-1," USONR, London, Eng.; January 1, 1959. 23 pp.

See especially section under Mathematical Sciences on the NPL symposium on mechanization of thought processes.

[209] Jones, R. W., "Models, analogues and homologues," *Regelungstechnik*, pp. 326–329; 1957. (In German.)

[210] Jouvet, M. "Neurophysiological Mechanisms of Learning: Final Technical Report," USAF Office of Scientific Res., Lyons University Lyons, France, Rept. No. TR-59-77; June, 1959. 35 pp.

While not computer oriented, this is an excellent background article on neurophysiological mechanisms of learning.

Kalaba, R., See [45], [46].

[211] Kamentsky, L. A., "Pattern and character recognition systems —picture processing by nets of neuron like elements," *Proc. WJCC*, San Francisco, Calif., March 3–5, 1959; pp. 304–309.

Describes an approach to the solution of pattern recognition that may be characterized as spatial operations by neuron-like elements. A model of a simplified neuron net was built and is discussed here.

——, See [196], [197].

[212] Kattsoff, L. O., "Brain, thinking and machines," *Methodos*, vol. 6, pp. 279–286; 1954.

A philosophical and technological discussion of the learning-machine problem.

Keller, J. M., See [57].

Kelley, J. L., See [167].

[213] Kemeny, J. G., "Man viewed as a machine," *Sci. Am.*, vol. 192, pp. 59–67; April, 1955.

Concludes that machines have not yet imitated the human brain's method of storing and recovering information. A discussion of the question, "Can a machine think?" is given with a description of the Turing machine. Many analogies of the machine and the human organism are given.

[214] Kesel, B., and A. De Lucia, "Information retrieval and language translation machines," *Behavioral Sci.*, vol. 4, p. 248; July, 1959.

Discusses programs for logical computations dealing with mechanical translation and information retrieval.

[215] Kilburn, T., R. L. Grimsdale and F. H. Sumner, "Experiments in machine learning and thinking," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France, June 15–20, 1959, pp. 303–309; 1960.

Experiments using the Manchester University computers to demonstrate machine learning and thinking are shown. A digital computer has been successfully programmed to generate its own programs which must satisfy certain given criteria.

[216] Kirsch, R. A., L. C. Ray, L. Cahn and G. H. Urban, "Experiments in processing pictorial information with a digital computer," *Proc. EJCC*, Washington, D. C., December 9–13; 1957; pp. 221–230.

A number of interesting experiments were performed on a digital computer, indicating that much could be accomplished through the use of a small number of very elementary operations, such as local averaging and spatial differentiation. The primary aim was to investigate the learning process, not the basic procedures of recognition or detection.

Kitor, A. I., See [407].

[217] Kleene, S. C., "Representation of Events in Nerve Nets and Finite Automata," in "Automata Studies," C. E. Shannon and J. McCarthy Eds., Princeton University Press, Princeton, N. J., pp. 3–41; 1956.

Discusses McCulloch-Pitts neuron nets and other representable stimuli for the purpose of discovering what events can be called "regular" and could be simulated in a finite automata.

[218] ——, "Representation of Events in Nerve Nets and Finite Automata," RAND Corp., Santa Monica, Calif., Rept. No. RM-704; December 15, 1951. 101 pp.

[219] Kobrinskii, A. E., M. G. Breido, V. S. Garfinkel, A. J. Sysin, M. L. Tzetlin and J. S. Yakobson, "A bioelectric control system," *Doklady Akademiia Nauk SSSR*, vol. 117, pp. 78–80; 1957. (In Russian.)

[220] Kochen, M., "Group behavior of robots," *Computers and Automation*, vol. 6, pp. 16–21; March, 1957.

A mathematical extension of logic usually applied to single automata.

[221] Kocher, D. G., "Sensory Aids Program: Reading Machines," Mass. Inst. Tech. Res. Lab. of Electronics, Cambridge, Quart. Progress Rept. No. 51, pp. 121–122; October 15, 1958.

In connection with the sensory aids program, a study was made of previous reading machines for the blind, and also of proposed methods and devices that might enable blind persons to read printed text. As a result of this study, it was concluded that efforts along this line should be continued.

[222] Koppel, H., "Digital computer plays NIM," *Electronics*, vol. 25, pp. 155–157; November, 1952.

Computer is used to play NIM, a pinball-type game.

[223] Kowalczyk, J., "Can a machine think?" *Wiez*, vol. 2, pp. 99–105; April, 1959. (In Polish.)

[224] Kremyanskii, V. I., "Certain Peculiarities of Organisms as a 'System' from the Point of View of Physics, Cybernetics, and Biology," Joint Publications Research Service, Washington, D. C., Rept. No. JPRS: 1356-N; 59-13302; March 17, 1959. 21 pp. (Translation of *Voprosy Filosofii*, vol. 8, pp. 97–107; 1958.)

[225] Krementulo, Iu. V., "Cybernetic 'turtle' 'Tortilla-2'," *Avtomatkya* (*Automation*), vol. 2, pp. 81–87; 1959. (In Russian.)

[226] Kuepfmueller, K., "Informationsverarbeitung durch den menschen," *Nachrichtentech. Z.*, vol 12, pp. 68–74; February, 1959. (In German.)

Human information processing; discussion of research carried out so far concerning the speed of human information processing, control actions and the role of human nerve nets and cells involved.

[227] Kunzfield, J., "An artificial dog," *Olomouc, Czechoslovak Republic. Vysoka skola Pedagogicka. Sbornik, Prirodni Vedy*, vol. 4, pp. 39–42; 1958. (In Czech.)

[228] Lashley, K. S., "Brain Mechanisms and Intelligence," University of Chicago Press, Chicago, Ill.; 1929.

[229] ——, "Cerebral mechanisms in behavior," in "Hixon Symposium," John Wiley and Sons, Inc., New York, N. Y., pp. 112–146; 1951.

[230] Latil, P. de, "Pensée Artificielle: Introduction à la Cybérnetique (The Artificial Thought: Introduction to Cybernetics), Gallimard, Paris; France; 1953. 332 pp. (In French.)

[231] ——, "Thinking by Machine, a Study of Cybernetics," Houghton Mifflin Co., Boston, Mass.; 1957. (Translated by Y. M. Golla.)

Emphasizes machine intellectual endeavor. A thorough coverage of recent concepts such as models of man-machine organisms, teaching machines, learning machines, etc. This is an idea book which affords excellent material for study and thought.

Lauria, F., See [59].

[232] Lebedev, S., "Machine that counts and translates," *Otvety na Voprosy Trudiashchkhsia* (*Replies to Workers Questions*), vol. 72, pp. 45–50; 1956. (In Russian.)

[233] Lee, R. J., "Generalization of learning in a machine," in "Preprints of Papers" from the 14th Natl. Meeting of the Association for Computing Machinery, at Mass. Inst. Tech., Cambridge, Association for Computing Machinery, New York, N. Y., pp. 21–24; 1959.

An artificial cerebral cortex which incorporates machine learning in general form is described. The scheme offers promise for new lines of basic research in self-programming data processing.

[234] Leeuw, K. de, E. F. Moore, C. E. Shannon, and N. Shapiro, "Computability by probabilistic machines," in "Automata Studies," C. E. Shannon and J. McCarthy, Eds., Princeton University Press, Princeton, N. J., pp. 183–212; 1956.

Considers the following question: Is there anything that can be done by a machine with a random element but not by a deterministic machine?

[235] Le Lionnais, F., "L'imitation de la Pensée Creatrice par les Machines," Université de Paris, Paris, France; 1957. 31 pp.

A popular lecture in which the author discusses various mechanical simulations of adaptive and intelligent behavior such as Grey-Walter's tortoise and the vocoder, Audrey and mechanical translation of languages.

[236] Lenneberg, E. H., "A probabilistic approach to language learning," *Behavioral Sci.*, vol. 2, pp. 1–12; January, 1957.

Experiments on language learning by using models for the acquisition of word meaning. The applications of such probabilistic approach should guide the computer person, although this article is not computer oriented.

Leondes, C. T., See [277], [278].

[237] Licklider, J. C., "Man-computer symbiosis," IRE Trans. on Human Factors in Electronics, vol. HFE-1, pp. 4–11; March, 1960.

Man-computer symbiosis is an expected development in cooperative interaction between men and electronic computers. It will involve very close coupling between the human and the electronic members of the partnership. The main aims are 1) to let computers facilitate formulative thinking as they now facilitate the solution of formulated problems, and 2) to enable men and computers to cooperate in making decisions.

Lijima, T., See [467].

[238] Linkovskii, G. B., "On the brain as a system of autonomous 'memory,'" *Biofizika*, vol. 3, pp. 385–390; 1958.

The thesis is proposed and developed that the brain from the mathematical point of view is an "assemblage of auto-oscillatory systems with feedback." These systems are described by means of "functional equations with lagging argument." The action of these systems is electrochemical. On the basis of the foregoing, the question of human memory is examined and a mathematical theory of dynamic memory is developed. Analysis demonstrates the existence of two systems of autonomous memory; one of absolute memory, another of relative memory. The brain is held to possess dynamic relative memory; although in certain parts of the brain purely static memory may exist.

[239] Liverant, S., "Intelligence, Time for a Change," University of New Mexico, Albuquerque, Rept. No. AFOSR TN-58-1111; 1958. 23 pp.

[240] Lofgren, L., "Automata of high complexity and methods of increasing their reliability by redundancy," *Proc. Ier Congres Internatl. de Cybernetique*. Assoc. Internatl. de Cybernetique, Namur, Belgium, pp. 493–511; 1948.

Surveys work of Turing and von Neumann on the question of complex machinery. States that present automation is not so much directed toward imitation of a human being, but rather toward attempts to imitate certain very specific features of man with more reliability.

[241] Lorente, G., "Remarques sur une theorie base de l'apprentissage," *Cybernetica*, vol. 2, pp. 127–135; February, 1959. (In French.)

Discusses the complex learning machine of Grey-Walter, Pask, and others.

[242] Lyapunov, A. A., "Cybernetics and its Future," Joint Publication Research Service, Washington, D. C., Tech. Repts. PB 141 270 T, JPRS (NY)-1-372; October, 1958. 7 pp. (Translation of *Moskovskiy Propagandist*, vol. 1, pp. 39–73; 1958.)

The following topics are discussed in connection with the future of cybernetics: Structural principal of control of manufacturing processes; processing of information; storing of information and how this may be accomplished.

[243] ——, "Mathematical problems in cybernetics," in "Cybernetics in the USSR," Joint Publications Research Service, Washington, D. C., Rept. No. JPRS 876-D, pp. 10–18; August 14, 1959. (Translation of *Izvestiia Vysshikh Uchebnykh Zavedeniia i Matematika* [*News of the Higher Educational Institutions and Mathematics*], vol. 5, pp. 166–174; June, 1958.)

The problem is basically a question of how and to what extent human thought can be described algorithmically.

[244] ——, "Some general problems in cybernetics," in "Cybernetics in the USSR." Joint Publications Research Service, Washington, D. C., Rept. No. JPRS 876-D, pp. 23–41; August 14, 1959. (Translation of "Problemy Kibernetiki [Problems in Cybernetics]," vol. 1, pp. 5–23; 1958.)

Covers the problem of interrelation of computer capability and human thought. This is linked to his belief that to solve this problem, the human processes of thought must be described "algorithmically."

——, See [407].

[245] MacCallum, D. M. and J. B. Smith, "Mechanized reasoning. Logical computers and their design," *Electronic Engrg.*, vol. 23, pp. 126–133; April, 1951.

[246] McCarthy, J., "The inversion of functions defined by Turing machines," in "Automata Studies," C. E. Shannon and J. McCarthy, Eds., Princeton University Press, Princeton, N. J., pp. 177–181; 1956.

Discusses the inversion of function as defined by Turing machines. The example given is of a well-defined problem.

[247] ——, "Programs with common sense," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 1, pp. 75–84; 1959.

Discusses programming computers to solve problems which require a high degree of intelligence in humans. Certain elementary verbal reasoning processes so simple that they can be carried out by any person have yet to be simulated by machine.

[248] ——, "Recursive functions of symbolic expressions and their computation by machine," *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 184–195; April, 1960.

Explanation of the LISP system which was developed for the IBM 704 computer by the Artificial Intelligence group at Massachusetts Institute of Technology is presented.

[249] ——, "Recursive Functions of Symbolic Expressions and their Computation by Machine," Mass. Inst. Tech., Res. Lab. of Electronics, Cambridge, Quart. Progress Rept. 53, pp. 124–152; April 15, 1959.

LISP programming system was designed to facilitate experiments with a proposed system called the Advice Taker, whereby a machine could be instructed to handle declarative as well as imperative sentences and could exhibit "common sense."

[250] —— and M. L. Minsky, "Artificial Intelligence, Research Objectives," Mass. Inst. Tech., Res. Lab. of Electronics, Cambridge, Quart. Progress Rept. 52, p. 129; January 15, 1959.

The purpose of this work is to investigate ways of making machines solve problems that are usually considered to require intelligence.

[251] ——, ——, and N. Rochester, "The LISP Programming System," Mass. Inst. Tech. Res. Lab. of Electronics, Cambridge, Quart. Progress Rept. 53, pp. 122–124; April 15, 1959.

The purpose of the programming system described here, called LISP, is to facilitate programming manipulations of symbolic expressions. This is part of a study of artificial intelligence at Massachusetts Institute of Technology.

[252] MacCarthy, R. A., "Electronic principles in brain design," *J. Irish Med. Assoc.*, vol. 37, pp. 325–329; November, 1955.

[253] McCulloch, W. S., "Agathe tyche, of nervous nets—the lucky reckoners," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 2, pp. 613–625; 1959.

The neurons, their excitations and inhibitions, are realistic and there remains sufficient redundancy for statistical control of growth to produce the synopsis of these stable, reliable, and flexible nets.

[254] ——, "The brain as a computing machine," *Elec. Engrg.*, vol. 68, pp. 492–497; June, 1949.

The author, a medical doctor, employs electrical engineering terminology to liken the brain to a digital computing machine consisting of ten billion relays called neurons. To carry the analogy further, the performance of the brain is governed by inverse feedback; subsidiary networks secure invarients, or ideas; predictive filters enable us to move toward the place where the object will be when we get there; and complicated servomechanisms enable us to act with facility and precision.

[255] ——, "Machines that think and want," *Comparative Psychological Monographs*, vol. 20, pp. 39–50; January, 1950.

The neuron is likened to a telegraphic relay which when tripped by a signal emits another signal. These are the atoms of the molecular events of the brain. All any neuron can signal to another neuron is that it was tripped. Mechanical processes such as negative feedback are related to neural function in purposive behavior. We should look to the synapse for the crucial changes which underlie learning.

[256] ——, "Three of von Neumann's Biological Questions," Mass. Inst. Tech., Res. Lab. of Electronics, Cambridge, Quart. Progress Rept., pp. 129–138; October 15, 1957.

Comments on John von Neumann's studies of components and connections in accounting for the steadiness and flexibility of behavior. His "Toward a probabilistic logic" states the problem of securing reliable performance from unreliable components, but his solution requires better relays than he could expect in brains. This article produces answers to questions of logical stability under common shift of threshold and of securing reliable performance from unreliable components without losing the flexibility of functions computed by nets composed of them.

[257] ——, "Toward some circuitry of ethical robots or an observational science of the genesis of social evaluation in the mind-like behavior of artifacts," *Acta Biotheoretica* (Leiden), vol. 11, pp. 147–156; 1956.

Modern knowledge of servo systems and computers makes it possible to specify a circuit which can indicate the rules in a game of chess when they are given only ostensibly; such circuits have a social value in the sense that it is shared by the player.

[258] —— and J. Pfeiffer, "Of digital computers called brains," *Sci. Monthly*, vol. 69, pp. 368–376; December, 1959.

Discusses relation between the principles, structures, functioning and malfunctioning of digital computers and the brain.

[259] —— and W. A. Pitts, "A logical calculus of the ideas imminent in nervous activity," *Bull. Math. Biophysics*, vol. 5, pp. 115–133; December, 1943.

The relationships that are used in symbolic logic can be realized in automata. This article proposes a model of a neuron net whose structure is a reflection of the logical propositions describing the activity of the net.

[260] ——, O. E. Oetlinger, O. H. Schmitt, and N. Rochester, "Design of machines to simulate behavior of human brain: Symposium," IRE Trans. on Electronic Computers, vol. EC-5, pp. 240–255; December, 1956.

Gives four short statements about what is being done in the use of computers to simulate the human brain. Of specific interest is the theoretical discussion about how the computer might be used to simulate the brain both by output and by duplicating the neural structure itself.

——, See [271], [337].

McDonald, H. S., See [105], [106].

[261] McGee, R. C., "Omnicode, a common language programming system," *Automatic Coding*, pp. 57–70; April, 1957.

[262] MacKay, D. M., "Calculating machines and human thought," *Nature*, vol. 167, pp. 432–434; March 17, 1951.

Discussion of a colloquium on "*Les Machines a Calculer et la Pensée Humaine*," organized by Institute Blaise-Pascal, January 8–13, 1958, at Paris.

The intention was to bring about an exchange of ideas between those concerned with the mechanical imitation of certain aspects of human thought and the nervous system.

[263] ——, "The epistomological problem for automata," in "Automata Studies," C. E. Shannon and J. McCarthy, Eds., Princeton University Press, Princeton, N. J., pp. 235–251; 1956.

Could an automaton develop and symbolize for itself a new concept as the occasion might arise? The aim in this paper is to distinguish between two broad lines by which a solution might be sought and then to discuss that phase which seems to answer the above question.

[264] ——, "Generators of information," in "Communication Theory," W. Jackson, Ed., Butterworths Scientific Publications, London, Eng., pp. 475–485; 1953.

[265] ——, "Mentality in machines, III," *Proc. Aristotelian Soc. for the Systematic Study of Phylosophie, Supplement*, London, Eng., vol. 26, pp. 61–86; 1952.

Illustrates and discusses the thesis that physical processes and mechanisms now known enable us in principle to mechanize a stochastic process of any complexity specifiable in the definition of a finite test. Discusses purpose, imagination, choice and volition and the status of these concepts as exemplified in the activity of an artifact.

[266] ——, "Mindlike behavior in artifacts," *Brit. J. Phil. Sci.*, vol. 2, pp. 105–121; August, 1951.

Discusses three points: Can an artifact be made to show behavioral characteristics of an organism; how closely in principle could the behavior of an artifact parallel the mind; on what philosophical issues do these possibilities rest?

[267] ——, "On comparing the brain with machines," *Am. Scientist*, vol. 42, pp. 261–268; June, 1954.

Discusses three points: How far does the brain resemble existing mechanisms? How far would it be possible to imitate human behavior with a suitably designed artifact? How far is it possible to envisage a model that would imitate human behavior and also work internally on the same principles as the brain?

[268] ——, "On the combination of digital and analogue computing techniques in the design of analytical engines," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 1, pp. 55–65; 1959.

This published version of a paper written in 1949 for private circulation is of interest historically.

[269] ——, "Operational aspects of intellect," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 1, pp. 37–52; 1959.

Concerns the theoretical problems of securing and evaluating intelligence in artificial organisms—particularly to distinguish calculation from intellect.

[270] ——, "Supra-logical behavior in automata," *Acta Psychol.*, vol. 11, pp. 204–205; 1955. (Abstract only.)

[271] ——, and W. S. McCulloch, "The limiting informational capacity of the neuronal link," *Bull. Math. Biophysics*, vol. 14, pp. 127–135; June, 1952.

[272] McKnight, J. D., "Mathematical Neurophysiology," Lockheed Aircraft Corp., Sunnyvale, Calif., Rept. No. TM LMSD 288038; October, 1959. 6 pp.

Summarizes some mathematical concepts of neurons.

[273] McPherson, R. R., "Systematic learning," Proc. IRE (Correspondence), vol. 44, p. 1054; August, 1956.

Discusses an idea of W. P. Tanner, Jr. pertaining to learning and system design. The actual behavior of the original system is called systemic learning.

[274] Mandelbrot. B., "An informational theory of the structure of language based upon the theory of the statistical matching of messages and coding," *Proc. Symp. on Applications of Communications Theory*, London, Eng., 1952, Butterworths Scientific Publications, London; 1953.

This is a feasibility study of the cost of communications.

[275] ——, "Théories du comportment, une définition de la cybernétique. Applications dans la languistiques," *Revue Générale des Sciences Appliquées*, vol. 62, pp. 278–294; September–October, 1955. (In French.)

[276] Mansberg, H. P., "Automatic particle and bacterial colony counter," *Science*, vol. 126, pp. 823–827; October 25, 1957.

Pattern-detection methods would be useful in biology where searches must be made for particular cell structures, bacteria, or viruses among a large number of microphotographs.

[277] Margolis, M. and C. T. Leondes, "On the Philosophy of Adaptive Control for Process Adaptive Systems," AF, Office of Scientific Res., Washington, D. C., Rept. No. TN-59-1199; January, 1960. 13 pp.

This paper describes a very general approach to the design of process adaptive systems utilizing the learning model.

[278] ——, ——, "On the Theory of Adaptive Control Systems, the Learning Model Approach," AF, Office of Scientific Res., Washington, D. C., Rept. No. TN-59-1200; October, 1959. 24 pp.

This paper describes the learning model which furnishes the parameters to the computing circuits of the programmer. The programmer then computes the values of the parameters of the controller and sets these parameters. For those varying systems where the learning model approach is applicable, the control laws can be applied as in classical control theory of stationary systems.

[279] Martens, H. H., "Two notes on machine 'learning'," *Information and Control*, vol. 2, pp. 364–379; December, 1959.

Generalizing from the examples given of a learning machine, the notion of L automation is introduced via a formal, behavioristic definition, in an attempt to give an abstract characterization of machine learning. A solution to the design problem for a general class of L automation is presented.

[280] Massachusetts Institute of Technology, Lincoln Lab., Div. 5, "Information Processing, Quarterly Progress Report 1," AF, Cambridge Res. Ctr., Mass., Rept. No. TN-59-1004; March 15, 1959. 107 pp.

See especially the section on pattern recognition, MAUDE, learning, and computers.

Mathews, M. V., See [105], [106].

[281] Mattson, R., "A Self-Organizing Binary System," Lockheed Aircraft Corp., Missiles and Space Div., Sunnyvale, Calif., Rept. No. LMSD 288029; September, 1959. (Various pagings.)

A self-organizing logical system may be thought of as two systems. One is a network of adjustable logical devices which receive binary input combinations and operate on these to produce a binary output. The other is a system for determining what adjustments should be made in the logical devices in order to produce a desired logical function.

[282] Mays, W., "Cybernetic models and thought processes," *Proc. Ier Congres Internatl. de Cybernetique (1st Internatl. Congress on Cybernetics)*, Assoc. Internatle. de Cybernetique, Namur, Belgium, pp. 103–110; 1958.

Discusses application of logical and probabilistic techniques to neurological and thought activities. Machines have been constructed for the specific purpose of simulating animal and human behavior. Views the efforts in this field and gives a perspective of the amount of work done.

[283] ——, "Mindlike behavior in artefacts and the concept of mind," *Brit. J. Phil. Science*, vol. 3, pp. 191–193; August, 1952.

[284] —— and D. G. Prinz, "A relay machine for the demonstration of symbolic logic," *Nature*, vol. 165, pp. 197–198; February 4, 1950.

Mc——: Names beginning with *Mc* are listed as though spelled *Mac*.

[285] Medvedev, I. T., "On a class of events representable in a finite automaton," in "Automata Studies" (Russian edition), A. A. Lyapunov Ed., Russian supplement to C. E. Shannon and J. McCarthy, Publishing Agency for Foreign Literature Press, Moscow, USSR; 1956. (This is a Russian article appearing as a final supplement to the Russian translation from English of "Automata Studies." Also issued as Mass. Inst. Tech. Lincoln Lab. Group Rept. 34-73; June 30, 1958.)

[286] Metelka, J., "The course of learning and forgetting in an electronic model," *Olomouc, Czechoslovak Republic, Vysoka Skola Pedagogicka, Sbornik, Prirodni Vedy*, vol. 4, pp. 33–38; 1958. (In Czech.)

[287] Mezentsev, V., "Thinking machines," *Stroitel' (The Builder)*, vol. 2, pp. 30–31; September, 1958. (In Russian.)

[288] Miles, T. R., "On the difference between men and machines," *Brit. J. Phil. Sci.*, vol. 7, pp. 277–292; February, 1957.

Author discusses how the concepts "man," "machine," "mind," and "consciousness" would operate if it ever happened that a *homo mechanisma* were constructed.

[289] Miller, G. A., "A note on the remarkable memory of man," IRE Trans. on Electronic Computers, vol. EC-6, pp. 194–195; September, 1957.

Miller, J. G., See [207].

[290] Minsky, M. L., "Heuristic Aspects of the Artificial Intelligence Problem," Mass. Inst. Tech., Lincoln Lab., Lexington, Group Rept. 34-55; December, 1956.

[291] ——, "Neural-Analog Networks and the Brain-Model Problem," Ph.D. dissertation, Princeton University, Princeton, N. J.; 1954. (Unpublished.)

[292] ——, "Some methods of artificial intelligence and heuristic programming," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 1, pp. 3–29; 1959.

Attempts to discuss and organize a number of ideas concerning the programming of machines to work on problems for which the designer does not have the solution. Particular attention is given to pattern recognition, learning, and models of problem solving.

[293] ——, "Some universal elements for finite automata," in "Automata Studies," C. E. Shannon and J. McCarthy Eds., Princeton University Press, Princeton, N. J., pp. 117–128; 1956.

It is shown that a certain category of elements are universal in the sense that one can assemble such elements into machines that can realize arbitrary functions within reasonable restrictions.

——, See [250], [251].

[294] Mobell, G., "World brains ponder mechanization of thought processes," *Automation and Automatic Equipment News*, vol. 4, pp. 929–934; January, 1959.

Discusses the Symposium on the Mechanization of Thought Processes held at Teddington, England.

[295] Moiseyev, K., "Man and the 'Thinking' Machine," Joint Publications Research Service, Washington, D. C., Rept. No. JPRS 2200-N; February 8, 1960. 15 pp. (In English from Russian.)

[296] Moles, A. A., "Principes d'incertitude de la perception et machines philosophiques," *Cybernetica*, vol. 2, pp. 51–58; 1959. (In French.)

Discusses the theories of communication and information in relation to the work of experimenters such as Grey-Walter and W. Ross Ashby.

[297] ——, "Le role de la cybernétique dans le dévelopement de la psychophysiologie," *Revue Générale des Sciences*, vol. 57, nos. 11, 12, pp. 253–261; 1950. (In French.)

[298] Mooers, C., "Machines for Information Retrieval, Learning and Translation," Zator Co., Cambridge, Mass., *Tech. Bull. 74;* 1952.

[299] Moore, E. F., "Gedanken-experiments on sequential machines," in "Automata Studies," C. E. Shannon and J. McCarthy Ed., Princeton University Press, Princeton, N. J., pp. 129–156; 1956.

This paper is concerned with finite automata and reports on what conclusions may be drawn about the internal conditions of a finite machine. The behavior of this machine is strictly deterministic.

——, See [234].

[300] Morozov, A., "Thinking machine," *Rabotnitsa (The Woman Worker)*, vol. 34, pp. 15–16; October, 1956. (In Russian.)

Mosteller, F. See [74]–[77].

Mugglin, M. G., See [429].

[301] Mukhin, I. S., "Syntax patterns in English studied by electronic computer," *Computers and Automation*, vol. 6, pp. 15–16; July, 1957.

[302] Mullin, A. A., "Some mathematical aspects of the analysis and synthesis of biological computers," in "The Realization of Biological Computers," University of Illinois, Elect. Engrg. Res. Lab., Urbana, 15, Quart. Progress Rept. 4, pp. 1–18; January 15, 1959.

[303] Murray, A. E., "A review of the Perceptron program," *Proc. Natl. Electronics Conf.*, vol. 15, pp. 346–356; 1959.

"Perceptron" is the class name for a family of pattern-recognition machines. They can learn to discriminate several categories. After exposure to a few samples in a category, such a machine tends to recognize spontaneously or classify correctly a new sample. This paper reviews principal conclusions from past work and indicates some plans for the future.

Napalkov, A. V., See [60]–[62].

[304] Naumann, H., "Es gibt kein elektronengehirn (There is no electronic brain)," *Orion*, vol. 14, pp. 57–63; January, 1959. (In German.)

[305] Neisser, U., "A Preliminary Study of Human Pattern Recognition," Mass. Inst. Tech., Lincoln Lab., Lexington, Group Rept. 34-75; September 12, 1958. 15 pp.

A method is discussed for the study of human recognition of complex and language-like patterns. The findings suggest tentatively that pattern recognition proceeds upward from the components of the pattern rather than across from one high-order unit to the next.

[306] ——, "A Theory of Cognitive Processes," Mass. Inst. Tech., Lincoln Lab., Lexington, Group Rept. 54-19; February 23, 1960. 15 pp.

A theory of cognitive processes is presented which maintains that the recognition of complex patterns depends on the prior recognition of simpler features of the same stimulus compounds which, in turn, depends on still earlier recognition of still more primitive features; cognition was concluded to be a hierarchically acquired process.

[307] Nerode, A., "Linear automaton transformations," *Proc. Am. Math. Soc.*, vol. 9, pp. 541–544; August, 1958.

[308] Newell, A., "The chess machine: An example of dealing with a complex task by adaptation," *Proc. WJCC*, Los Angeles, Calif., March 1–3, 1955; pp. 103–108.

The example given illustrates "adaptation" as case for learning by machine. The problems outlined are very general. No attempt is made to go into detail, but rather to present a fund of ideas for further exploration.

[309] ——, "On Programming a Highly Parallel Machine to be an Intelligent Technician," RAND Corp., Santa Monica, Calif., Rept. No. P-1946; 1960. 16 pp.

[310] ——, "Problem Solving in Humans and Computers," RAND Corp., Santa Monica, Calif., Rept. No. P-987; December 7, 1956. 12 pp.

Compares thinking processes of modern electronic computers and human beings. Presents research in psychology using electronic computers to reach an understanding of human mental processes.

[311] ——, "The Processes of Creative Thinking," RAND Corp., Santa Monica, Calif., Rept. No. P-1320; January 28, 1959. 82 pp.

Presented at a Symposium on Creative Thinking, University of Colorado, Boulder, May 16, 1958. Concludes that there is no need for a theory of problem solving. Summarizes what has been learned about problem solving by simulating certain human problem solving processes with digital computers.

[312] —— and F. M. Tonge, "An introduction to information processing language V," *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 205–211; April, 1960.

An explanation of the IPL System for programming of the simulation of cognitive processes.

[313] ——, J. C. Shaw, and H. A. Simon, "Chess-playing programs and problem of complexity," *IBM J. Res. & Dev.*, vol. 2, pp. 320–335; October, 1958.

Discusses: development of digital-computer programs that play chess; work of Shannon, Turing, Los Alamos group, Bernstein and other authors; relation of chess problem studies to understanding and construction of complex intelligent machines.

[314] ——, ——, ——, "Elements of a theory of human problem solving," *Psychol. Rev.*, vol. 65, pp. 151–166; May, 1958.

Describes theory of problem solving in terms of information processes amenable for use in a digital computer. The postulates are: a control system consisting of a number of memories, which contain symbolized information and are interconnected by various ordering relations; a number of primitive information processes which operate on information in the memories; and a perfectly definite set of rules for combining these processes into whole programs of processing.

[315] ——, ——, ——, "Empirical explorations of the logic theory machine," *Proc. WJCC*, Los Angeles, Calif., February 26–28, 1957; pp. 218–230.

[316] ——, ——, ——, "A general problem-solving program for a computer," *Computers and Automation*, vol. 8, pp. 10–16; July, 1959.

Discusses theory of problem solving. Deals with a program for a digital computer called General Problem Solver I, which is part of an investigation into processes involved in intelligent, adaptive, creative behavior.

[317] ——, ——, ——, "Report on a general problem-solving program," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France, June 15–20, 1959, pp. 256–264; 1960.

Reports on the General Problem Solving Program I. The approach is synthetic—to construct computer programs that can solve problems requiring intelligence and adaptation and to discover which varieties of these problems can be matched with human problem solving.

[318] ——, ——, ——, "Report on a General Problem-Solving Program," RAND Corp., Santa Monica, Calif., Rept. No. P-1584; February 9, 1959. 27 pp. (Presented at Internatl. Conf. on Information Processing, Paris, France; June 15–20, 1959.)

The approach is synthetic: to construct computer program to solve problems requiring intelligence and adaptation.

[319] —— and H. A. Simon, "The Simulation of Human Thought," RAND Corp., Santa Monica, Calif., Rept. No. P-1734; June 22, 1959. 41 pp.

Describes a method of studying human problem solving, gives an application of the method, and indicates theory of problem solving that emerges. Method consists of constructing a theory of central processes in the form of a program, demonstrating sufficiency of theory to produce problem solving behavior by realizing it in a computer, and testing theory against human processes by comparing the trace generated by the program with protocol of a human subject.

——, See [398], [404].

[320] Newman, E. A., "Machines that try to think," *Control*, vol. 1, pp. 294–295; December, 1958.

A rewrite of the Symposium on the Mechanization of Thought Processes held in Teddington, England.

[321] Newman, E. B., "Men and Information: A Psychologist's View," Harvard University, Cambridge, Mass., Rept. No. AFCRC TR 60-51; 1959. 21 pp.

An article of information on men and machines with special emphasis on the cybernetics of neurophysical mechanisms and the analogy in machines.

Niehaus, U. K., See [468].

[322] Nikolaev, O., "On machines which think," *Molodaia Gvardiia; Literaturno-Khudozh-Estvennyi i Obshchestvenno Politicheskaii Zhurnal (Young Guard; Literary and Sociopolitical Magazine)*, vol. 8, pp. 219–221; August, 1959. (In Russian.)

This is a book review of "Faster Than Thought," by N. Kobrinskii and V. Pekelis.

North, J. H., See [136].

[323] Notterman, J. M. and R. Trumbull, "Note on self-regulating systems and stress," *Behavioral Sci.*, vol. 4, pp. 324–327; October, 1959.

If the components of performance of self-regulating systems can be precisely defined for the gross behavior of an organism responding to its environment, the study of the effects of stress can be made precise. It would be useful to know whether nonspecific responses, sometimes interpreted as anxiety symptoms, are attempts of the organisms to get more feedback from environment.

Oetlinger, O. E., See [260].

[324] Oettinger, A. C., "Programming a digital computer to learn," *Phil. Mag.*, vol. 43, pp. 1243–1263; December, 1952.

Digital computers are equipped with a memory in which past information can be stored and modified by present actions. The computer when suitably programmed can, at least in principle, be trained.

[325] ——, "Simple learning by a digital computer," *Proc. Assoc. Comp. Mach.*, Toronto, Ont., Can., September 8–10, 1952; pp. 55–61.

Okumura, Y., See [467].

Onesto, N., See [59].

[326] Pahl, P. M. and L. Johnson, "Pattern recognition in an electronic reader," *The Trend*, vol. 11, pp. 16–21; July, 1959.

To improve upon such mechanism, reference is made to the analysis of the human nervous system.

[327] Pask, G., "Artificial organisms," *General Systems; Yearbook of the Society for General System Research*, vol. 4, pp. 151–170; 1959.

This paper is concerned with Richard L. Meierx's experimental organism called *Artoga* which is a decision-making network involving human beings as active elements and the possibility of simulating the pattern of behavior by a computer program. It has developed into a discussion of how such organisms could operate and be adaptive.

[328] ——, "Organic control and the cybernetic method," *Cybernetica*, vol. 1, pp. 155–173; 1958.

Applies cybernetic method of industrial processes. The solution of industrial processes implies a mode of behavior rather than a definite point, and having achieved this, the control mechanism is equivalent to a manager—replacement and reinforcement of men by machines.

[329] ——, "Physical analogues to the growth of a concept," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 2, pp. 879–922; 1959.

Discusses the circumstances in which a machine is said to think, and describes a mechanical process which is said to correspond to concept formation.

[330] ——, "Tomorrow's control systems can learn from experience," *Automation Progress*, vol. 4, pp. 43–45; February, 1959.

Computers, we are always told, cannot do anything for which they do not have detailed instructions. But experimental machines have been described and constructed which exhibit certain advanced features of the brain, such as, learning by experience and spontaneously building up of a model of the relevant parts of the surroundings.

Pattishall, E. G., See [36].

[331] Paycha, F., "Medical diagnosis and cybernetics," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 2, pp. 637–659; 1959.

The role of logic in medicine and medical diagnosis is explained in relation to cybernetics. Suggests a scheme for automatic diagnosis and prognosis.

[332] Pedoe, D., "The Gentle Art of Mathematics," The Macmillan Co., New York, N. Y.; 1959. 143 pp.

A short, interesting book written for the intelligent layman. There are nine chapters on such subjects as: mathematical games, stochastic choice, automatic thinking, etc.

[333] Perlis, A. J. and C. Thornton, "Symbol manipulation by threaded lists," *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 195–204; April, 1960.

In the field of artificial intelligence, many of the most interesting problems do not lend themselves to solutions in the automatic pro-

gramming systems now in use. Examples are given of manipulation of threaded lists from elementary algebra. The threaded-lists principle is similar to that of human problem solving.

[334] ——, ——, "Symbol Manipulation by Threaded Lists," Carnegie Inst. Tech., Computation Ctr., Pittsburgh, Pa., Rept. No. TR-1; March 1, 1960. 29 pp.

[335] —— and J. W. Smith, "A mathematical language computer," *Automatic Coding*, pp. 87–102; April, 1957.

Pfeiffer, J., See [258].

[336] Pierce, C. S., "Logical machines," *Am. J. Psychol.*, vol. 1, pp. 165–170; November, 1887.

States that a machine could be constructed which would solve the logic of relations with a large number of terms. Of historical interest.

Pike, W. S., See [498].

[337] Pitts, W. and W. S. McCulloch, "How we know universals, the perception of auditory and visual forms," *Bull. Math. Biophysics*, vol. 9, pp. 127–147; September, 1947.

Two neural mechanisms are described which can exhibit recognition of forms. Both are independent of small perturbations at synapses and are referred to particular regions of the nervous system. These are now referred to as the Pitts-McCulloch neurons.

——, See [259].

[338] Platt, J. R., "Amplification aspects of biological response and mental activity." *Am. Scientist*, vol. 44, pp. 180-196; April, 1956.

An evaluation of three disciplines that have aided our understanding of biological behavior and mental activity. These are: feedback theory; information theory; and the theory of games. The author here presents the fourth, amplifier theory. It is based on a parallel between biological responses and electronic processes.

[339] Polonsky, J., "Essai d'interprétation du fonctionnement des cellules vivantes dans le cadre de la cybernétique quantique," *Ann. Radioélectricité*, vol. 13, pp. 346–370; October, 1958. (In French.)

Essay on interpretation of functioning of living cells based on quantum cybernetics; discussion of hypothesis that all living cells act from physical point of view, as microcybernetic electromagnetic system governed by quantum laws and comprising generators of information.

[340] Polya, G., "Mathematics and plausible reasoning," in "Patterns of Plausible Inference," Princeton University Press, Princeton, N. J.; 1954.

Outlines heuristic principles which enable us to arrive at judgments, yet Polya states that there can never be a machine which can perform such plausible inferences. This must mean that the future of such "learning machines" is not bright. However, most workers concerned in the area find him a useful source of methods for use in self-organizing systems.

[341] Poplavko, Iu., "Calculating machines in cybernetics" *Dnipro (Dniepro)*, vol. 32, pp. 133-137; October, 1958. (In Russian.)

[342] Porter, A. and P. K. T. Vaswani, "The optimization of logical goal-seeking procedures," *J. Electronics and Control*, vol. 6, pp. 168–185; February, 1959.

The property of decision making is basic in all types of control systems, and the goal-seeking characteristics of such systems can be considered as a direct consequence of this capability. There is a difficulty in optimizing results of systems which are based on two-valued logical statements.

[343] Pospelov, V., "Thinking machines," *Sovetskii Soiuz (Soviet Union)*, vol. 3, pp. 43–45; 1959. (In Russian.)

Post, G. See [41].

[344] Precker, J. A., "Toward a theoretical brain-model," *J. Personality*, vol. 22, pp. 310–326; September, 1954.

Considers three systems: Krech, dynamic systems; Hebb, phase sequences; Wiener and Coutu, cybernetics and tendency in situation. A brain model mainly following the theory of Hebb is constructed.

[345] Pribram, K. "On the neurology of thinking," *Behavioral Sci.*, vol. 4, pp. 265–287; October, 1959.

A basic article on the neurology of thinking, which is basically active uncertainty or trial-and-error learning. Includes an excellent bibliography on neurology.

[346] Prinz, D. G., "Robot chess," *Research*, vol. 5, pp. 261–266; June, 1952.

Describes how an electronic computer can be programmed to play chess.

——, See [284].

[347] Prywes, N. S., "Multi-List Organized Associative Memory for Moore Intelligent Machines," Moore School of Elec. Engrg., University of Pennsylvania, Philadelphia; 1959. 39 pp.

An organization for an information retrieval system of improved efficiency is presented and analyzed.

[348] Rabin, M. O. and D. Scott, "Finite automata and their decision problems," *IBM J. Res. & Dev.*, vol. 3, pp. 114–125; April, 1959.

Finite automata may be considered as a subclass of Turing machines having a finite number of internal states and finite tapes. In this they are similar in concept to nerve nets. The abstract theory of finite automata, divided into one-tape, one-way, one-tape, and multi-tape, are discussed.

[349] Rapoport, A., "Contribution to the probabilistic theory of neural nets," *Bull. Math. Biophysics*, vol. 12, pp. 109–121; June, 1950.

[350] ——, "Technological models of the nervous system," *ETC Rev. General Semantics*, vol. 11, pp. 272–283; 1954.

[351] ——, "Technological models of the nervous system," *Methodos*, vol. 7, pp. 131–146; 1955.

Presents the present (1955) state of the art of learning machines with reference to the use of models of the nerve structure.

[352] ——, "Technologic models of nervous system," *Psychiatric Res. Rept. No. 2*, pp. 119–131; December, 1955.

[353] —— and A. Shimbel, "Mathematical biophysics, cybernetics and general semantics," *ETC Rev. General Semantics*, vol. 6, pp. 145–159; 1949.

Rise of unifying sciences such as mathematical physics, biological chemistry, physiological psychiatry, and others are discussed. These fields pursue programs of research aimed at the discovery of significant invariants underlying the functioning of the nervous system. Research on a mathematical theory of the nervous system and development of electronic computers proceeds along parallel lines.

[354] Rashevsky, N., "Mathematical biophysics of abstraction and logical thinking," *Bull. Math. Biophysics*, vol. 7, pp. 133–148; September, 1945.

[355] ——, "The neural mechanism of logical thinking," *Bull. Math. Biophysics*, vol. 8, pp. 29–40; March, 1946.

[356] Rausch, F., "Self-checking methods in electronic calculating machines," *Elektron. Rundschau*, vol. 13, pp. 206–210; June, 1959. (In German.)

After a brief discussion of programmed checks, the question of built-in checking of circuits is examined. As examples, the self-checking diagrams are given for various error-detecting and correcting circuits.

Ray, L. C., See [216].

[357] Raymond, F. H., "L'automatique des informations," Masson et Cie, Paris, France; 1957. 187 pp.

A French book on computers and information handling on a philosophical level. The book ends with a short discussion on "thinking" by computers.

[358] Reinberg, M. G., "Dumaiushchie mashiny (Thinking machines)," Detgiz, Moscow, USSR; 1957. 125 pp. (In Russian.)

[359] ——, "Electrons control, compute and think," *Izobretate'l i Ratsionalizator (Inventor and Efficiency Promoter)*, vol. 10, pp. 39–41; October, 1958. (In Russian.)

[360] Reitman, W. R., "Heuristic programs, computer simulation tion and higher mental processes," *Behavioral Sci.*, vol. 4, pp. 330–335; October, 1959.

Reviews the work of Shaw, Simon and Newell in the development of heuristic computer programs. Heuristic programs are concerned with simulating the higher mental processes of humans—thinking and problem solving. Emphasis has been on research and not on the computer.

[361] Remington Rand Univac, "Project Lightning," *Quart. Rept. 1*, St. Paul, Minn.; December 1, 1958–February 28, 1959. 61 pp.

Reports on a major effort at Remington Rand to study the mathematical and logical techniques for exploiting the special properties of this ultra-high-speed computer. Section four, pages 50–61 is, especially interesting since it is on mathematical and logical research.

[362] Rice, R. J., "Simulation of nerve cells by electro-chemical methods," *Brit. Commun. and Electronics*, vol. 6, pp. 846–848; December, 1959.

After a brief statement of the general requirements for a model neuron in simulated nerve nets, the operation and applications of an electro-chemical model neuron for use in cybernetic and automata models are described.

[363] Richards, P. I., "Machines which can learn," *Am. Scientist*, vol. 39, pp. 711–716; October, 1952.

This is an extension of an article by Weinberg who proposed that a computational chess-playing machine be given a large memory to add up experiences and thereby improve its ability to function. Richards' exploratory article here wishes to start this machine with no information whatever except for a large memory and then let it add up experiences and optimalize them by using its intelligence.

[364] ——, "On game-learning machines," *Sci. Monthly*, vol. 74, pp. 201–205; April, 1959.

Can one conceive of a machine that has absolutely no built-in knowledge but does have an intelligence and an ability to learn almost any game through experience alone? A list of suggestions for the design of such a machine is presented, with likely imperfections and analogies in human behavior pointed out.

[365] Richens, R. H., "Tigris and Euphrates—a comparison between human and machine translation," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 1, pp. 281–302; 1959.

Roberts, N. deV., See [51], [52].

[366] Robles, E. R., "System of 3 'neurodes' in retina, similarity to electronic brain," *Archivos de la Sociedad Oftalmologica Hispano-Americana*, vol. 11, pp. 1508–1512; December, 1951. (In Spanish.)

[367] Rochester, N., S. H. Goldberg, and D. J. Edwards, "Machine Manipulation of Algebraic Expressions," Mass. Inst. Tech. Res. Lab. of Electronics, Cambridge, Quart. Progress Rept. 55, pp. 132–134; October 15, 1959.

A listed-processing computer programming language called LISP is used to program the IBM 704 to analyze linear bilateral electric network whose components have literal or numerical values. This work was produced as part of Massachusetts Institute of Technology's artificial intelligence study. During the evaluation of characteristics for complicated electric networks, the machine can produce algebraic expressions that are far too complicated for any human being to understand.

[368] ——, J. H. Holland, L. H. Haibt, and W. L. Duda, "Test on a cell assembly theory of the action of the brain, using a large digital computer," IRE Trans. on Information Theory, vol. IT-2, pp. 80–93; September, 1956.

Presents a detailed description of a series of computer simulation studies designed to test the Hebb-Milner theories of brain action. In general, this paper presents much detailed program logic regarding characteristics and interconnections of simulated neurons. May be interpreted as limited positive support for Hebb and perhaps Rosenblatt.

——, See [146], [251], [260].

[369] Romain, J., "Information et cybernétique," *Cybernetica*, vol. 2, pp. 22–50; 1959. (In French.)

Information and cybernetics, analysis of basic notions of information theory such as information and signal coding, etc., in order to relate them to the wider framework of physiology and cybernetics.

[370] Rosenblatt, F., "The design of an intelligent automaton," *Research Trends*, vol. 6, no. 2, pp. 1–7; Summer, 1958.

Reports the basic ideas behind a type of thinking machine which supposedly can recognize forms, learn the correct response to them, and communicate results.

[371] ——, "On the Convergence of Reinforcement Procedures in Simple Perceptrons," Cornell Aeronautical Lab., Buffalo, N. Y., Rept. No. VG-1196-G-4; February 15, 1960. 66 pp.

Earlier reports have described a number of theoretical nerve nets called Perceptrons and have attempted to analyze their ability to associate responses to stimuli, using a variety of training procedures and memory functions. This report presents mathematical foundations rather than heuristic arguments.

[372] ——, "Perceptron simulation experiments," Proc. IRE, vol. 48, pp. 301–309; March, 1960.

An experimental simulation program, which has been in progress at the Cornell Aeronautical Laboratory since 1957, is described. This program uses the IBM 704 computer to simulate perceptual learning, recognition, and spontaneous classification of visual stimuli in the Perceptron, a theoretical brain model which has been described elsewhere. This paper includes a brief review of the organization of simple Perceptrons, and theoretically predicted performance curves are persented.

[373] ——, "The Perceptron: a probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, pp. 386–407; March, 1958.

A theory is developed for a hypothetical nervous system called a Perceptron. The theory serves as a bridge between biophysics and psychology. It is impossible to predict learning curves from neurological variables. The quantitative statistical approach is fruitful in understanding the organization of cognitive systems.

[374] ——, "The Perceptron: a theory of statistical separability in cognitive systems," Cornell Aeronautical Lab., Buffalo, N. Y., Rept. No. VG-1196-G-1; January, 1958. 268 pp.

A basic document, discussing the organization of the Perceptron, and its statistical theory.

[375] ——, "Perceptron Simulation Experiments (Project PARA)," Cornell Aeronautical Lab., Buffalo, N. Y., Rept. No. VG-1196-G-3; June, 1959, 34 pp.

This paper is concerned with a report of digital simulation experiments which have been carried out on the Perceptron, using the IBM 704 computer at the Cornell Aeronautical Laboratory. These experiments are intended to demonstrate the performance of particular systems in typical environment situations, free from any approximations which have been used in the previously published mathematical analysis.

[376] ——, "A probabilistic model for visual perception," *Proc. 15th Internatl. Congress of Psychol.*, 1957, North-Holland Publishing Co., Amsterdam, The Netherlands, pp. 296–297; 1959.

[377] ——, "Tables of Q-Functions for Two Perceptron Models (Project PARA)," Cornell Aeronautical Lab., Buffalo, N. Y., Rept. No. VG-1196-G-6; May, 1960. 144 pp.

[378] ——, "Two theorems of statistical separability in the Perceptron," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 1, pp. 421–456; 1959.

Discusses the theoretical brain model developed at Cornell Aeronautical Laboratory as a probabilistic system, capable of learning.

[379] ——, "Two Theorems of Statistical Separability in the Perceptron (Project PARA)," Cornell Aeronautical Lab., Buffalo, N. Y., Rept. No. VG-1196-G-2; September 1, 1958. 42 pp.

[380] Ross, T., "Machines that think," *Sci. Am.*, vol. 148, pp. 206–208; April, 1933.

Assuming that thought is a mechanical process, this author describes a few thinking devices.

[381] ——, "The synthesis of intelligence—its implications," *Psychol. Rev.*, vol. 45, pp. 185–189; 1938.

[382] Russell, G., "Learning Machines and Adaptive Control Mechanisms," British Ministry of Supply, Malvern, Eng., R. R. E. Memo No. 1369; 1957.

[383] Ruzic, N. P., "Automata," *Industrial Res.*, vol. 1, pp. 47–59; Spring, 1959.

A popular article on automata and an advanced computer able to program itself.

[384] Samuel, A. L., "Some studies in machine learning using the game of checkers," *IBM J. Res. & Dev.*, vol. 3, pp. 210–229; July, 1959.

Investigation in some detail of machine learning procedure using the game of checkers has verified that a computer can be programmed so that it will learn to play a better game of checkers than can be played by the designer of the program.

[385] Schaefer, E., "The human memory as an information store," *Elektron. Rundschau*, vol. 14, pp. 79–84; March, 1960. (In German.)

A review paper summarizing much of what is known about the human brain from the functional point of view and contrasting its capacity with mechanical devices.

Schmitt, O. H., See [260].

Scott, D., See [348].

[386] Selfridge, O. G., "Pandemonium: A Paradigm for Learning," Mass. Inst. Tech., Lincoln Lab., Lexington; April 29, 1960. 12 pp.

A model is proposed of a process which it is hoped can adaptively improve itself to handle certain pattern-recognition problems which cannot be adequately specified in advance. The basic idea behind the

model is the notion of parallel processing. Examples of patterns taken from some sets of them are presented to the model, each time informing the model which pattern was presented.

[387] ——, "Pandemonium: a paradigm for learning," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 1, pp. 513–526; 1959.

Proposes a model which will adaptively improve itself to handle certain recognition problems not specified in advance.

[388] ——, "Pattern recognition and learning," in "Information Theory," Colin Cherry, Ed., Butterworth Scientific Publications, London, Eng., pp. 345–353; 1956.

The author suggests that there are two ways to overcome a machine's want of a useful descriptive language. One could provide the machine with an *a priori* set of aphorisms or teach the machine a language in chess by having it play simpler games.

[389] ——, "Pattern recognition and learning," *Methodos*, vol. 8, pp. 163–176; 1956.

Comments on the synthesis of the learning process and the application of pattern-recognition principles to the mechanical imitation of learning.

[390] ——, "Pattern recognition and modern computers," *Proc. WJCC*, Los Angeles, Calif., March 1–3, 1955; pp. 91–93.

Pattern recognition is defined as extraction of significant features from a background of irrelevant detail. This involves a matter of selection by the computer.

[391] Sergrove, J. A., "Automatic inspection, the anatomy of conscious machines," *Proc. Ier Congres Internatl, de Cybernetique* (*1st Internatl. Congress on Cybernetics*), Assoc. Internatl. de Cybernetique, Namur, Belgium pp. 597–619; 1958.

In an automatic factory, the inspection process must be automated, that is, endowed with the ability to decide, or (as these authors say) consciousness.

[392] Shannon, C. E., "A chess-playing machine," *Sci. Am.*, vol. 182, pp. 48–51; February, 1950.

Reports on a computer which plays games and which in turn raises the question of whether or not a machine can think.

[393] ——, "Computers and automata," *Methodos*, vol. 6, pp. 115–130; 1954.

Describes logic machines, game-playing and learning machines. Compares the computer and the brain, Turing's work and von Neumann's models of self-reproducing machines.

[394] ——, "Game-playing machines," *J. Franklin Inst.*, vol. 260, pp. 447–453; December, 1955.

Traces the historical development of game-playing devices culminating with the electronic computer.

[395] ——, "Presentation of a maze solving machine," *Trans. 8th Conf. on Cybernetics*, pp. 173–180; 1951.

[396] ——, "Programming a computer for playing chess," *Phil. Mag.*, vol. 41, pp. 356–375; March, 1950.

One of the first chess-playing formulations.

[397] ——, "A universal Turing machine with two internal states," in "Automata Studies," C. E. Shannon and J. McCarthy Eds., Princeton University Press, Princeton, N. J., pp. 157–166; 1956.

Shows that a universal Turing machine can be constructed using one tape and having only two internal states. Also shows that it is impossible to do this with one internal state. A construction is given for a universal Turing machine with only two tape symbols.

——, See [234].

Shapiro, N., See [234].

[398] Shaw, J. C., A. Newell, H. A. Simon, and T. O. Ellis, "A command structure for complex information processing," *Proc. WJCC*, Los Angeles, Calif. May 6–8, 1958; pp. 119–128.

Concerns a linguistic structure for man-machines that will produce greater efficiency in processing computer information.

——, See [313]–[318].

[399] Shepherdson, J. C., "The reduction of two-way automata to one-way automata," *IBM J. Res. & Dev.*, vol. 3, pp. 198–200; April, 1959.

Gives direct proof that two-way finite automata are allowed to move in both directions along which their input tapes are equal to one-way automata.

[400] Sherman, H., "A quasi-topological method for the recognition of line patterns," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France, June 15–20, 1959, pp. 232–238; 1960.

The ability of the human being to recognize line patterns such as handwriting and printing has long been a goal for machine emulation. Such machine is herein proposed.

[401] Shimbel, A., "Some elementary considerations of neural models," *Bull. Math. Biophysics*, vol. 14, pp. 67–71; March, 1952.

——, See [353].

Shrieder, Yu. A., See [62].

[402] Shultz, G. L., "Use of IBM 704 in simulation of speech-recognition systems," *Proc. EJCC*, Washington, D. C., December 9–13, 1957; pp. 214–218.

[403] Simmons, R. F., "Anticipated Developments in Machine Literature Processing in the Next Decade," System Development Corp., Santa Monica, Calif., Rept. No. SP-129; March 10, 1960. 13 pp.

Briefly reviews V. Bush's concept of the general-purpose literature retrieval machine, the synthesizing computer "synthex." Compares current progress of self-organizing systems and their existing related hardware to the requirements of these machines. Concludes that an era of nontrivial thinking machine may be at hand in the next decade.

[404] Simon, H. A. and A. Newell, "Heuristic problem solving: the next advance in operations research," *Operations Res.*, vol. 6, pp. 1–10; January–February, 1958.

Learning machines and their closeness to human logic are discussed.

——, See [313]–[319], [398].

[405] Singh, B. and J. Chandy, "Cybernetics and neurons," *Indian J. Med. Sci.*, vol. 7, pp. 33–46; January, 1953.

[406] Smilga, V., "Is an automatic chess player machine possible?" *Shakmaty v SSSR* (*Chess in the USSR*), vol. 33, pp. 176–177; June, 1956. (In Russian.)

Smith, J. B., See [245].

Smith, J. W., See [335].

[407] Sobolev, S. L., A. I. Kitor, and A. A. Lyapunov, "The Basic Features of Cybernetics," Library of Congress, Photoduplication Service, Washington, D. C., Rept. No. JPRS/DC-L-15; February 25, 1958. 24 pp. (Translation from Russian of *Voprosy Filosofii*, vol. 4, pp. 136–147; 1955.)

Discusses the theory of information as a statistical theory of processing and transmitting concepts. Also discusses the theory of an automatic electronic computing machine as a self-organizing logical process similar to the processes of human thinking.

[408] Sokolovskii, Iu. I., "Kibernetika nastoiashchego i budushchego; o 'razumnykh' mashinakh, iskusstvennykh organakh chuvstv, avtomatichenskom perevode knig, mathematicheskoi logike i fiziologii nervnoi deiatel'nosti (Cybernetics in the present and in the future, 'thinking' machines, artificial organs of feeling, automatic translation of books, mathematical logic and physiology of nervous activity)," *Khar'kovskoe knizhnoe izd-vo*; 1959. 190 pp.

[409] Solodovnikov, V. V., "Cybernetics," Joint Publications Research Service, Washington, D. C., Rept. No. PB 141 269T, JPRS(NY)-L-371; October 5, 1958. 10 pp. (Translation of *Nauka i Zhizn'*, vol. 6, pp. 18–22; 1957.)

The following topics are discussed to explain the fundamental concepts of cybernetics: origin and scientific end of cybernetics; concept of information; feedback concept; the living organism and the information machine; technical cybernetics.

[410] Solomonoff, R. J., "An inductive inference machine," 1957 IRE National Convention Record, pt. 2, pp. 56–62.

Programs have been written for a computer to perform simple mathematics with success, but less success has been achieved for more complex problems.

[411] ——, "On Machines to Learn to Translate Languages and Retrieve Information," Zator Co., Cambridge, Mass., *Progress Rept. ZTB-134;* October, 1959. 17 pp. (Presented at the Internatl. Conf. for Standards on a Common Language for Machine Searching and Translation, sponsored by Western Reserve University and RAND Corp., Cleveland, Ohio, September 6–12, 1959.)

[412] Somenzi, V., "Can induction be mechanized?" in "Information Theory," Colin Cherry, Ed., Butterworths Scientific Publications, London, Eng., pp. 226–230; 1956.

A discussion of the possibility of mechanizing induction and reasoning.

[413] ——, "Can induction be mechanized?" *Methodos*, vol. 7, pp. 147–151; 1955.
Mechanical imitation of deduction is considered since the earliest experiments with logical computers. The author tries to show how a machine could predict future values of a physical magnitude on the basis of physical laws of induction for past values.

[414] Spence, K. W., "Mathematical theories of learning," *J. Gen. Psychol.*, vol. 49, pp. 283–291; October, 1953.
Postulates that mathematical learning theories fall into two groups: quantitative theories based on neurophysiological foundations, and those providing a set of constructs and postulates concerning basic processes underlying all learning situations.

[415] Spiegelthal, E. S., "Computing educated guesses," *Proc. WJCC*, San Francisco, Calif., March 3–5, 1959; pp. 70–73.
Attempts to automate one area hitherto considered an exclusive human domain: error detecting and error correcting.

[416] Spilsbury, R. J., "Mentality in machines, II," *Proc. Aristotelian Soc. for the Systematic Study of Phylosophie, Supplement (London)*, vol. 26, pp. 27–60; 1952.
After discussing feedback, learning, imagination, and imitation in artifacts, author concludes that there are apparently insoluble problems (not merely technological) which in our present information-state rule out the possibility of designing an artifact which will possess quite ordinary human qualities and capacities, let alone creative powers and insight.

[417] Sprick, W. and K. Ganzhorn, "An analogous method for pattern recognition by following the boundary," *Proc. Internat. Conf. on Information Processing*, UNESCO, Paris, France, June 15–20, 1959, pp. 238–244; 1960.
This paper summarizes the design, realization and results of the work which has been done in the field of the recognition of numerals in the last few years.

[418] Steinbuch, K., "Automatic symbol recognition," *Nachrichtentech Z.*, vol. 11, pp. 210–219; April, 1958. (In German.)
Study of possible techniques for achieving direct and infallible conversion of print into electric signals.

[419] ——, "Automatic symbol recognition," *Nachrichtentech Z.*, vol. 11, pp. 237–244; May, 1958. (In German.)
The continuation and conclusion of an earlier article.

[420] ——, "Introductory speech," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France, June 15–20, 1959, pp. 223–226; 1960.
A general introductory article on pattern recognition and machine learning.

[421] ——, "Lernende automaten," *Elektron. Rechenanlagen*, vol. 1, pp. 112–118, 172–174; August–November, 1959. (In German.)
Similar to Friedberg's learning machines in English.

[422] Stephens, J. M., "A mechanical explanation of the law of effect," *Am. J. Psychol.*, vol. 41, pp. 422–431; July, 1929.
This paper attempts to present a description of a learning machine and a theory of learning.

[423] Stevens, M. E., "A machine model of recall," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France, June, 15–20, 1959, pp. 309–315; 1960.
A model of certain logical recall operations involving both pattern recognition and a limited degree of machine learning is presented.

[424] Stewart, D. J., "A notation for logical nets," *Cybernetica*, vol. 2, pp. 59–69; 1959.
Suggests an improved form of notation for logical nets, owing a great deal to the forerunners, men such as McCulloch and Pitts. Gives brief description of background and main assumptions of logical net theory.

[425] Stroud, J., "The psychological moment in perception," *Trans. Sixth Conf. on Cybernetics*, New York, N. Y., March 24–25, 1949, Josiah Macy, Jr. Foundation, pp. 27–63; 1950.
Discusses the place of humans in a servo-system and feedback in the psychological system. The emphasis here is on the understanding of psychological process.

[426] Sukharebskii, L., "Brains and cybernetics," *Praha-Moskva (Praha, Czech)*, vol. 8, pp. 974–976; November, 1958. (Translated from Russian into Czechoslovakian.)

[427] Sullivan, D. L., "A Short Bibliography of the Literature Relating to Learning Machines and Other Automata," AF Cambridge Res. Ctr., Bedford, Mass., Tech. Memo Rept. No. ERD-CRRB-TM-58-128; September, 1958.

Sumner, F. H., See [215].

Sutherland, I., See [429].

[428] Sutherland, N. S., "Stimulus analysing mechanism," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 2, pp. 577–601; 1959.
Problems of stimulus analyzing mechanism in organisms are discussed.

[429] Sutherland, W. R., M. G. Mugglin, and I. Sutherland, "An electro-mechanical model of simple animals," *Computers and Automation*, vol. 7, pp. 6–8; February 23, 1958.
An electromechanical model is presented so that its processes may be studied and improved for control applications.

[430] Swallow, R. and P. Weston, "On the Design of Artificial Nerve Nets," in The Realization of Biological Computers," University of Illinois, Elect. Engrg. Lab., Urbana, *Quart. Progress Rept. 4*, pp. 34–46; January 15, 1959.
Discusses design of nerve nets for what is called artificial intelligence. Contrasts the two opposing schools, logical and biological.

Sward, G. L., See [118].

[431] Swartz, P., "Perspectives in psychology: VI. A note on the computing machine analogy in psychology," *Psychol. Record*, vol. 8, pp. 53–56; 1958.
The uncritical use of analogy in psychology and sciences in general is discussed. Comments by R. R. Oppenheimer, J. W. Krutch, and J. R. Kantor relative to this problem are examined.

Sysin, A. J., See [219].

Takahashi, S., See [467].

[432] Tarjan, R., "Neuronal automata," *Cybernetica*, vol. 1, pp. 189–196; 1958.
Since the advent of electronic digital computers, there has been much discussion at various levels about the possibility of constructing machines with animal-like behavior, or machines that think. The thesis here is that at present machines can be built to imitate behavior of life, but cannot think.

[433] Taylor, W. K., "Automatic control by visual signals," *Proc. Symp. Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 2, pp. 843–855; 1959.
Electrical networks for performing the signal transformations and analog-digital conversion are described and analyzed. The characteristics of pattern recognition and reading machines synthesized by the new method are compared with those exhibited by the human operator. Presents a concrete description of pattern recognition logic.

[434] ——, "Automatic pattern recognition," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng. Her Majesty's Stationery Office, London, vol. 2, pp. 951–952; 1959. (A supplement to [433].)

Temperley, H. N. V., See [97].

[435] Teplov, L., "The Brain and the Machine," in "Cybernetics in the USSR," Joint Publications Research Service, Washington, D. C., Rept. No. JPRS 876-D, pp. 1–9; August 14, 1959. (Translation of *Teknika Molodezhi (Technology for Youth)*, vol. 2, pp. 5–6; 1959.)
This is a very popular Russian article on computer-brain comparison.

Thornton, C., See [333], [334].

[436] Tomkins, S. S., "La conscience et l'inconscient representes dans un modéle de l'être humain (The conscious and the unconscious represented in a model of the human being)," *Psychanalyse*, vol. 1, pp. 275–286; 1956. (In French.)
The relationship of conscious and unconscious activity in experience and behavior may be represented in a model of the human being based on information theory and cybernetics. Three major classes of motives are defined within such a model: addictions, formulas and inventions.

Tonge, F. M., See [312].

[437] Trimmer, J. D., "Instrumentation and cybernetics," *Sci. Monthly*, vol. 69, pp. 328–331; November, 1949.
Discusses the relationship of instrumentation, cybernetics, and system response. Suggests that the ultimate superinstrumentation

taking shape might be broadened as follows: observation-generation of information by instruments, sense organs, computing machines, etc.; communication-transfer of information by devices, man-machine systems, etc.; control-transfer and use of information by regulators and servos, etc.

Trumbull, R., See [323].

[438] Truxal, J. G., "Trends in adaptive control systems," *Proc. Natl. Electronics Conf.*, vol. 15, pp. 1–16; 1959.

An adaptive feedback control system is defined as a configuration in which the measurement of process dynamics or signal characteristics is utilized to adjust automatically the controller in an attempt to achieve optimum operation at all times. Although adaptivity is essentially a viewpoint in analysis and design, a variety of novel systems has resulted from the early years of research in adaptive systems. The ultimate goal is a feedback system possessing the adaptivity of a high-order learning mechanism.

[439] Turing, A. M., "Can a machine think?" in "The World of Mathematics," James R. Newman Ed., Simon and Schuster, Inc., New York, N. Y., vol. 4, pp. 2099–2123; 1956.

A general discussion of machines and thinking which ends by considering thinking and learning machines.

[440] ——, "Computable numbers with an application to entscheidungsproblem," *Proc. London Math. Soc.*, vol. 42, ser. 2, pp. 241–265; December 23, 1936.

Defines the Turing machines which have an infinite tape, permitting them to have more complicated behavior than the finite automata.

[441] ——, "Computing machinery and intelligence," *Methodos*, vol. 6, pp. 195–223; 1954. (Reprinted from [442].)

[442] ——, "Computing machinery and intelligence," *Mind*, vol. 59, pp. 433–460; July, 1950.

Many machines capable of taking over operations defined as thinking are possible, but digital computers can do some of these now and within 50 years will probably be able to equal the human mind. Suggests the construction of a learning machine having the approximate complexity of a child's mind and that could be educated.

[443] Tustin, A., "Cybernetics and all that," *J. Brit. IRE*, vol. 1, pp. 634–635; October, 1955.

[444] —— "Physical models and psychological processes," *J. IEE (London)*, vol. 6, pp. 201–204; April, 1960.

The brain is an electrical system and can be investigated by means of electrical instruments. The electrical engineer therefore has a special interest in the exploratory work being done by neurophysiologists. In this article, work is reviewed on the brain processes and recent pattern recognition mechanisms.

Tzetlin, M. L., See [219].

[445] Uhr, L., "Intelligence in computers: the psychology of perception in people and machines," *Behavioral Sci.*, vol. 5, pp. 177–182; April, 1960.

A long abstract reporting research and work done in the field of computers in the field of psychology and intelligent machines.

[446] ——, "Intelligence in computing machines: the psychology of perception in people and in machines," in "The Impact of Computers on Psychological Research," Philip A. Smith, Ed., System Development Corp., Santa Monica, Calif., Rept. No. SP-119; October 21, 1959. 30 pp.

Discusses the potential abilities and the quantity of human intelligence that can be programmed into a computer. Special reference to pattern recognition. Presented at the 1959 Annual Meeting of the American Psychological Association, Cincinnati, September 7, 1959.

[447] ——, "Latest methods for the conception and education of intelligent machines," *Behavioral Sci.*, vol. 4, pp. 248–251; July, 1959.

Account of papers given at the Western Joint Computer Conference, 1959, dealing with pattern recognition, information retrieval, mechanical translation and problem-solving and learning machines.

[448] ——, "Machine Perception of Forms by Means of Assessment and Recognition of Gestalts," The University of Michigan, Mental Health Res. Inst., Ann Arbor, Preprint No. 34; October, 1959.

Discusses perception of forms by computer, based on the work of Minsky, MacKay and Gelernter. This is based on a paper read at the meeting of the Association for Computing Machinery, September 10, 1959.

[449] ——, "Machine perception of printed and handwritten forms by means of procedures for assessing and recognizing Gestalts," in "Preprints of Papers," 14th Natl. Meeting of the Assoc. for Computing Machinery at Mass. Inst. Tech., Association for Computing Machinery, New York, N. Y., pp. 20–26; 1959.

Using a computer for machine perception and recognition of printed and handwritten language and forms.

[450] Unger, S. H., "A computer oriented toward spatial problems," PROC. IRE, vol. 46, pp. 1744–1750; October, 1958.

A general-purpose digital computer can, in principle, solve any well-defined problem. At many tasks, such as the solution of systems of linear equations, these machines are thousands of times faster than human beings. However, they are relatively inept at solving many problems where data is arranged naturally in a spatial form. For example, when it comes to playing chess or recognizing sophisticated patterns, present-day machines cannot match the designers.

[451] ——, "Pattern detection and recognition," PROC. IRE, vol. 47, pp. 1737–1752; October, 1959.

Describes how machines duplicate some of the remarkable feats which humans perform daily. The first is pattern detection, such as when a postal clerk deciphers a badly written address, and the second is pattern recognition, *e.g.*, the ability to recognize and read letters and picture characters, such as photos in a rogues' gallery line-up.

Urban, G. H., See [216].

[452] Uttley, A. M., "L'application des méthodes d'apprentissage par approximation successives aux commandes automatiques (The use of learning methods by successive approximations in automatic computer instructions)," *Automatisme*, vol. 4, pp. 18–37; January, 1959. (In French.)

[453] ——, "Conditional probability machines and conditioned reflexes," in "Automata Studies," C. E. Shannon and M. McCarthy, Eds., Princeton University Press, Princeton, N. J., pp. 253–275; 1956.

It is deduced that the principles of design of a machine whose reactions to stimuli are similarities are suggested between the structure of a machine and of the human nervous system.

[454] ——, "Conditional probability computing in a nervous system," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 1, pp. 119–147; 1959.

An expansion of Uttley's proposal for a conditional probability computer.

[455] ——, "The Conditional Probability of Signals in the Nervous System," Radar Res. Establ., Malvern, Worcestershire, Eng., Memorandum 1109; 1955.

Discusses recognition pattern by pattern rather than element by element in his concept of conditional probability machine.

[456] ——, "The design of conditional probability computers," *Information and Control*, vol. 2, pp. 1–24; April, 1959.

A special-purpose computer is described which calculates conditional probabilities. The input to the computer is a set of channels in either an active or inactive state. At any instant a particular set of channels will, in general, be active; the computer calculates the probability of activity in the other channels, based on what has happened in the past. Such a computer uses the principle of induction, and it can imitate many forms of animal learning.

[457] ——, "Information, machines, and brains," IRE TRANS. ON INFORMATION THEORY, no. PGIT-1, pp. 143–149; February, 1953.

Points out certain similarities of function, and of behavior between computing machines and animals, among them, storage, transformation of information, memory, etc.

[458] ——, "The probability of neural connections," *Proc. Roy. Soc. (London) B*, vol. 144, no. 916, pp. 229–240; September 27, 1955.

[459] ——, "Temporal and spatial patterns in a conditional probability machine," in "Automata Studies," C. E. Shannon and J. McCarthy Eds., Princeton University Press, Princeton, N. J., pp. 277–285; 1956.

Discusses the problem of pattern completion faced by a conditional probability machine for temporal and spatial patterns.

[460] ——, "A theory of the mechanism of learning based on the computation of conditional probabilities," *Proc. Ier Congrès Internatl. de Cybernetique (1st Internatl. Congress on Cybernetics)*, Assoc. Internatl. de Cybernetique, Paris, France, pp. 830–856; 1958.

Vaswani, P. K. T., See [342].

[461] Verdier, J., "Thought governs machines," *Vzashchitu Mira (In Defense of Peace)*, vols. 87/88, pp. 123–125; August–September, 1958. (In Russian.)

[462] Vernon, M. D., "The functions of schemata in perceiving," *Psychol. Rev.*, vol. 62, pp. 180–192; 1955.

[463] Von Foerster, H., "Quantum mechanical theory of memory," *Trans. 6th Conf. on Cybernetics*, New York, N. Y., March 24–25, 1949, Josiah Macy, Jr., Foundation, pp. 112–145; 1950.
Postulates a quantum mechanical theory of memory based on the mathematics of biological systems and feedback in cybernetics.

[464] ——, "Some aspects in the design of biological computers, appendix," in "The Realization of Biological Computers," University of Illinois, Elec. Engrg. Lab., Urbana, *Quart. Progress Rept. 4*, pp. 47–64; January 15, 1959. (Presented at the Second Internatl. Conf. on Cybernetics, Namur, Belgium, September 5, 1958.)
Discusses four aspects in design of biological computers: what we want to do; what we do not want to do; componentry needed; theoretical program.

[465] von Neumann, J., "The general and logical theory of automata," in "Cerebral Mechanisms in Behavior," L. A. Jefferies, Ed., John Wiley and Sons, Inc., New York, N. Y., pp. 1–41; 1951.
Further develops a theory of automata by Shannon, McCulloch, and Pitts in which is proposed a model of a neural net whose structure is a reflection of the logical propositions describing the activity of the net. They have shown that for any consistent set of such propositions a net can be constructed.

[466] ——, "Probabilistics logics and the synthesis of reliable organisms from unreliable components," in "Automata Studies," C. E. Shannon and J. McCarthy, Eds., Princeton University Press, Princeton, N. J., pp. 43–98; 1956.
Thesis is that error is treated as extraneous and misdirected in the human intellectual system. Author believes that this is false and that error should be treated by thermodynamical methods and be subject to more exhaustive study.

[467] Wada, H., S. Takahashi, T. Lijima, Y. Okumura, and K. Imota, "An electronic reading machine," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France, June 15–20, 1959, pp. 227–232; 1960.
The design of this machine is similar to that of the Solartron ERA with a diode matrix. It will recognize 72 characters and other symbols.

[468] Wall, R. E. and U. K. Niehaus, "Russian to English machine translation with simple logical processing," *Commun. and Electronics*, vol. 34, pp. 709–714; January, 1958.

[469] Walter, W. Grey-, "Imitation of mentality," *Nature* (London), vol. 177, pp. 684–685; April 14, 1958.

[470] ——, "Les machines à calculer et la pensée humaine," *Colloque International*, Centre National de Recherche Scientifique, Paris, France, vol. 12, pp. 407–421; 1951. (In French.)

[471] ——, "Studies on activity of the brain," *Trans. 10th Conf. on Cybernetics, Circular, Causal and Feedback Mechanisms in Biological and Social Systems*, Josiah Macy, Jr. Foundation, New York, N. Y., pp. 19–31; 1955.
A general article about cybernetics and the common analogy of the brain and its mechanical analogs. Culminates on a happy note about the future of machines.

——, See [33].

[472] Walton, A. "Conditioning illustrated by an automatic mechanical device," *Am. J. Psychol.*, vol. 42, pp. 110–114; 1930.
For classroom demonstration, a mechanical toy was created which would illustrate and exhibit a conditioned response to a substitute stimulus.

[473] Wang, H., "Proving theorems by pattern recognition—I," *Commun. Assoc. Comp Mach.*, vol. 3, pp. 220–234; April, 1960.
This paper concerns further work on proving theorems of Principia Mathematica with an IBM 704 reported in the author's paper, "Toward mechanical mathematics."

[474] ——, "Toward mechanical mathematics," *IBM J. Res. & Dev.*, vol. 4, pp. 28–35; January, 1960.
A new feature is suggested to replace essentially exhaustive methods by a study of the patterns according to which extensions involved in the search for a proof are continued. An IBM 704 program employing this feature is described in some detail.

——, See [71].

Ward, L. B., See [49].

[475] Ware, W. H., Ed., "Soviet Computer Technology—1959," RAND Corp., Santa Monica, Calif., Rept. No. RM-2541; March, 1960. 192 pp.
Contains a section on "applications," which discusses information retrieval, character recognition and theory of human memory; theory by Linkovsky, a Russian engineer and mathematician.

[476] Watson, A. J., "Some questions concerning the explanation of learning in animals," *Proc. Symp. on Mechanization of Thought Processes*, Natl. Physical Lab. Symp. No. 10, Teddington, Eng., Her Majesty's Stationery Office, London, vol. 2, pp. 693–720; 1959.
Concerns the question of learning in animals and the experimental evidence together with psychological theories which have been developed to answer them. Points to applications in machines which learn.

[477] Weber, C. O., "Homeostasis and servo-mechanisms for what?" *Psychol. Rev.*, vol. 56, pp. 234–236; July, 1949.
Begins with brief statements of the two related principles, *i.e.*, homeostasis and servo-mechanisms, then indicates and evaluates the ways in which homeostasis has been applied to mental processes and presents limitations of these concepts.

[478] Weinberg, M., "Mechanisms in neurosis," *Am. Scientist*, vol. 39, pp. 74–98; January, 1951.
Viewing the nervous system as a device of high order for reception and utilization of signals, some features of its design can be postulated. Another device, Shannon's chess-player, modified to be able to exhibit neurosis, is taken as an analog. By examining the nervous system as a problem in design, a pattern of reflection by which patterns can be recognized and learning can take place is illustrated.

Weston, P., See [430].

[479] White, G. W., "Electronic probability generators," *Rev. Sci. Instr.*, vol. 30, pp. 825–828; September, 1959.
Investigation of nonstationary random processes may arise in many models of control and in learning systems. To aid in studying this class of systems, an electronic probability generator (EPI) was built. This can electronically flip a coin at a maximum rate of 500 times a second. The probability of heads can be varied.

[480] ——, "Penny Matching Machines," General Electric Co., Schenectady, N. Y., Res. Lab. Rept. 59-RL-2296E; 1959.

[481] ——, "Penny matching machines," *Information and Control*, vol. 2, pp. 349–363; December, 1959.
A problem currently facing engineers is the design of machines capable of making decisions. This article contains a description of the strategies a machine or player might use in a simple penny matching game. Under certain conditions, optimum decisions can be performed with a single analog storage element.

[482] Wiener, N., "Speech, language and learning," *J. Acoust. Soc. Am.*, vol. 22, pp. 696–697; November, 1950.

[483] Wier, J. M., "A physical model of an abstract learning process," in "Preprints of Papers," 13th Natl. Meeting of the Assoc. for Computing Machinery at Mass. Inst. Tech., Association for Computing Machinery, New York, N. Y., pp. 43-1–43-2; 1958.
This learning model attempts to examine learning as a process independent of the physical mechanisms which normally carry out such activities; this is referred to as "psychological learning."

[484] Wilkes, M. V., "Can machines think?" Proc. IRE, vol. 41, pp. 1230–1234; October, 1953.

[485] Willis, D. G., "Plastic neurons as memory elements," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France, June 15–20, 1959, pp. 290–298; 1960.

[486] ——, "Plastic neurons as memory elements," 1959 IRE WESCON Convention Record, pt. 4, pp. 55–65.
Two demonstrations are presented showing the read-in and read-out properties of a system of 288 plastic neurons simulated on a digital computer. An expression is derived for the output of a plastic neuron as a function of its previous history.

[487] ——, "Plastic Neurons as Memory Elements," Lockheed Missiles and Space Div., Sunnyvale, Calif., Tech. Rept. LMSD-484-32; March, 1959. 26 pp.
This paper precisely defines a class of plastic neuron models, determines their information storage capacity, and shows the mechanism by which information can be read into and out of a system of such neurons on a random access basis. Appears to be compatible with the requirements of the human memory.

[488] Wisdom, J. O., "The hypothesis of cybernetics," *Brit. J. Phil. Sci.*, vol. 2, pp. 1–24; May, 1951.

An excellent overview of the hypothesis of cybernetics with special emphasis on feedback and adjustment in biological and machine systems.

[489] ——, "Mentality in machines, II," *Proc. Aristotelian Soc. for the Systematic Study of Phylosophie, Supplement* (London), vol. 26, pp. 1–26; 1952.

Starts from the hypothesis that adaptive behavior can be explained by feedback and other auxiliary mechanisms in the nervous system. Discusses the question as to whether in reproducing human activity in an artifact adequately, at a minimum it could want and like things, weigh eventualities, and have imagination. General conclusion is that mentality cannot be reproduced in an artifact without the discovery of some radically new type of mechanism.

[490] Wood, W. D. and B. L. Basore, "A Communication Channel with Learning," Dikewood Corp., Albuquerque, N. M., Tech. Rept. TN-1-1004; September, 1959. 18 pp.

A generalization of information theory to include learning is reported in this document. Assuming that the receiver can recognize the elements of the message set but is unfamiliar or inexperienced with their probabilities of occurrence, a modified "entropy" concept can be defined using subjective probabilities for measuring the elements of surprise to the receiver.

Wright, J. B., See [91].

[491] Wright, M. A., "Can machines be intelligent?" *Process Control and Automation*, vol. 6, pp. 2–6; January, 1959.

Wrightman, C. W., See [193].

[492] Wrigley, C., "Theory construction or fact-finding in a computer age?" *Behavioral Sci.*, vol. 5, pp. 183–186; April, 1960.

Discusses the effect of computers or psychological methods, and suggests that fact-finding may be more important than theorizing; computers make possible a much more intensive search for multivariable functional relationships and theory is therefore less important as a guide to what to focus on.

[493] Wyatt, J. K., "Prediction by computer," *Data Processing*, vol. 1, pp. 137–141; July–September, 1959.

Description with simplified flow chart of program, of how Glacier Metal Company Ltd., used a Ferranti Pegasus to simulate operations in a proposed new machine shop to give management a quantitative basis for decision making.

[494] Wyckoff, B., Jr., "A mathematical model and an electronic model for learning," *Psychol. Rev.*, vol. 61, pp. 89–97; March, 1954.

A robot device which can be confronted with learning problems is described. Its performance corresponds to the performance prescribed by a mathematical model.

[495] Young, D. A., "Automatic character recognition," *Electronic Engrg.*, vol. 32, pp. 2–10; January, 1960.

Following an introductory survey of some of the principal automatic character-recognition systems that are currently in either the production or research stage, the fundamental semantic features and limits of character patterns are discussed. It is possible to formulate definitions of character patterns that are particularly amenable to translation in terms of discriminatory logic.

[496] Zemanek, H., "La tortue de vienne et les autres travaux cybernétiques," *Proc. Ier Congrès Internatl. de Cybernétique* (*1st Internatl. Congress on Cybernetics*), Association Internatl. de Cybernetique, Paris, France, pp. 770–780; 1958.

[497] Zurabashvii, A. D., "O kibernetike i nekotorykh voprosakh psikhiatrii (On cybernetics and several problems in psychiatry)," *Zhurnal Neuropathologia Psikhiat.*, vol. 58, pp. 1264–1269; 1958.

With all the great insights and prospects which cybernetics offers, its limitations for psychiatry and psychoneurology are also pronounced. The human brain, though it has self-regulatory aspects, cybernetically definable, is essentially very different in its operations from those of automatic machines. This should never be forgotten.

[498] Zworykin, V. K., L. E. Flory, and W. S. Pike, "Letter reading machine," *Electronics*, vol. 22, pp. 80–86; June, 1949.

Print scanned by a flying spot reflects varying amounts of light to a multiplier phototube. Binary counters operate appropriate trigger circuits; magnetic recordings pronounce the letters. Designed as an aid for the blind, the equipment could be modified for use in connection with printing and communication systems.

Subject Index—The Simulation of Cognitive Processes

A Note on the System Requirements of a Digital Computer for the Manipulation of List Structures*

HERBERT GELERNTER†

Summary—The technique of programming within the framework of the so-called Newell-Shaw-Simon (NSS) associative list memory is currently the subject of much interest among workers in the area of advanced programming research. Unfortunately, committing a given program to list memory generally entails accepting a significant loss of speed and efficiency in information processing, so that the advantages accruing from the use of list memory must be carefully balanced against its weaknesses. This paper is concerned with the system requirements of a digital computer for which the use of list techniques is to be competitive with standard programming, so that the particular memory organization for a given problem may be chosen on the basis of suitability and ease of programming alone. A description of a list-processing 7090-type computer is appended in order to make our discussion concrete, although engineering and economic feasibility is not implied.

Introduction

THE past few years have witnessed an impressive expansion in the number of research efforts in the area of non-numerical data processing. Among the research workers immediately concerned, there is general agreement that a large number of such non-numerical problems are best programmed within the framework of the so-called Newell-Shaw-Simon associative list memory. (See Newell and Shaw [5]; Yngve [7]; Gelernter, Hansen, and Gerberich [2]; and McCarthy [4].)

In general, list storage is prescribed wherever the intermediate data generated by a program are unpredictable in their form, complexity, organization, and length. In such programs, arbitrary lists of information may or may not contain as data an arbitrary number of items or sublists. To allocate beforehand to each possible list a block of storage sufficient to contain some reasonable maximum amount of information would quickly exhaust all available fast-access storage as well as prescribe rigidly the organization of information within the lists. (Of the two, the latter is probably the more serious limitation for problems requiring symbol manipulation.) Program failure caused by some list exceeding its allotted space while most of the remainder of storage is empty could be expected to occur frequently. List-organized storage provides a convenient solution for these difficulties.

Programming for the problems of language translation, information retrieval, game playing, and machine intelligence, for example, fall within the category defined above. There is little doubt, too, that a number of real-time military problems could be dealt with most conveniently with list techniques, providing that efficient list-processing computers were available. Beyond these pure programming applications, the equivalent of an NSS list has found its way into the logical design of the Bull Gamma 60 computer to cope with the real-time queueing problem for instructions in a system where many functional units are operated simultaneously [1]. There is little wonder, then, that people have begun to think in terms of computers specifically designed for list processing (Shaw, *et al.* [6]), and, in fact, it is to this problem that the present paper is addressed.

The remarks below will reflect strongly the author's conviction that high-speed digital computers as they are currently constituted are far from obsolete for non-numerical data processing. Experience with the relatively abstract problem of theorem-proving in Euclidean plane geometry [3] indicates that the efficiency of such programs may often be vastly increased by occasionally computing the address of a quantity rather than searching for it on a list. Too, the arithmetization of certain logical operations sometimes offers further dividends in program efficiency, even granting the computer a full complement of logical commands. Finally, it seems clear that there is a vast area for exploration in the interaction of symbol-manipulating data-processing systems with number-manipulating ones. Our approach, then, will be to specify a complement of additional list and logical processes to be embodied within the hardware of an otherwise traditional high-speed electronic computer.

List Processing

A *list* is the fundamental assemblage of information in NSS storage. Each register of NSS memory is an element of at least one list; if it is not on some information list, then it is on the *list of available storage* (LAVST), which serves as a source of raw material for list formation processes, and as a sink for dissociated registers when information is destroyed. At the beginning of any program that is to make use of list-processing techniques, some fraction of high-speed memory must be assigned to NSS storage by placing a predetermined number of registers on LAVST.

In standard addressable memory, information is stored and retrieved by associating a given datum with a location in storage. At the lowest level of program interpretation, the name of that location is represented by a positive integer. In normal use, related information is distributed in memory in such a way that there exists some simply computable relation between the addresses

* Received by the PGEC, December 12, 1960; revised manuscript received, April 25, 1961.

† IBM Research Center, Yorktown Heights, N. Y.

associated with that information. For example, the relation *next* between data is commonly represented by the operation "add 1" between the associated addresses.

In list memory, information is stored and retrieved by virtue of its membership in a given information class, or set of classes. Within a class, a datum may further be associated with an ordinal number, or else with an identifying tag. At the machine level of interpretation, the name of a class, or list, is again a positive integer. The data on a given list may, of course, include the names of other lists.

It must be emphasized that necessity has often forced the incorporation of certain features of a list memory into the common garden variety of program. The introduction of the concept of NSS list memory merely systematizes and standardizes the use of these techniques, so that completely general list processes may be defined independent of any particular utilization of list structure.

In the NSS representation of a list memory, a given datum is placed in a given class either by storing it in the information field of one of a directed string of linked[1] list elements, or else by linking it to one such list element. In the particular representation of Gelernter, Hansen, and Gerberich [2] (henceforth GHG), the list elements, or *location words*, are linked by placing in the decrement field of a given location word the machine address of the succeeding word. The primary information field of a list element is the address field of the location word. A datum is linked to a list element by placing the machine name of that datum (*i.e.*, its address) in the information field. Since that datum can itself be a list, it is easy to see that a list can become an extremely complex, highly-connected network-like structure (Fig. 2 of GHG [2], for example), and is, therefore, especially well suited for representing such structures within a digital computer.

It is the purpose of this report to suggest the kind of hardware additions and modifications to a standard digital-computer system that would enable the computer to process information organized in list storage far more efficiently than is possible in conventionally designed machines. To clarify our discussion, the several points we wish to make will be illustrated by giving examples of suggested system modifications for the IBM 7090. Our examples, therefore, will reflect the word substructure of sign, prefix, decrement, tag, and address that is natural for that computer. We wish to point out, however, that engineering and economic feasibility for such modification of the 7090 is not to be construed from this report.

Indirect Addressing

An obvious requisite for any information-processing system is a means for extracting from their natural habitat the data to be manipulated. In the case of a list memory, the operation of information retrieval entails, in general, the extraction of the address of the required data from the address or decrement field of a word whose address must in turn be extracted · · · , etc. The foregoing process is clearly a generalization of the concept of indirect addressing which is already a feature of the 7090 computers. Our generalization requires that the indirect-addressing process be permitted to continue to an arbitrary depth; that the decrement and address fields of a word be treated symmetrically; and that the process be fully specified by the indirectly-addressed instruction, *independent of the contents of the intermediate registers traversed in the indirect-address chain.* The requirement that decrement and address fields be treated symmetrically means that at each stage of the indirect-address operation, the current effective address is to be replaceable by either the decrement or the address field of the word *at* the current effective address. Based upon experience acquired in using a Fortran-compiled list-processing language (described in detail in GHG [2]), it is reasonable to assert that hardware implementation of generalized indirect addressing would at least double the list-processing efficiency of a 7090-class computer.

Effective-Address Recovery

In current 700-series systems, the computed effective address obtained as a result of some address modification process (indexing, for example) is not directly accessible to the program. While this is a fact of little consequence for ordinary numerical computation, the ability to recover for further processing the terminus of a long indirect-address string would represent a substantial gain for a system designed to manipulate list structures. Provision for effective-address recovery becomes almost a necessity if one wishes to generalize the concept of "list element" to include units of more than one computer word.

Ideally, a special "effective address register" (EAR) should be provided. Two bits in the instruction format would then be associated with that register, one to indicate that the effective address computed for a given instruction is to be stored in EAR, the other to indicate that the contents of EAR are to be added to the address of a given instruction. A two-bit effective-address tag may contain sufficient information to allow the specification of two different modes for the latter process. In the first mode, EAR is added to the instruction address *before* modification; in the second mode, it is added to the computed effective address of that instruction.

The more immediate problem of providing for effective-address recovery in the 7090 could most economically be solved within the framework of the existing index registers. The particular index register chosen to serve as EAR would be designated by a special instruction. If the 7090 instruction format cannot be made to yield the two indicator bits necessary to specify the more flexible EAR processes described above, a single bit

[1] A register whose address is contained in a specified 15-bit field of a list element is said to be *linked* to that element.

could be used to signal the loading of EAR. The contents of EAR would then be extracted by indexing an instruction in the normal way.[2]

Field Logic

Most existing stored-program computers contain within their instruction repertoires a limited set of operations for manipulating specific subfields of the full fixed-length computer word, and indeed, some computers (STRETCH, for example) provide complete flexibility in field logic by allowing the programmer to address individual bits or fields anywhere within core storage, unrestrained by word boundaries. While such flexibility is always a boon for programming complex information processes, the field logic requirements of a list-manipulating computer are rather less demanding. Particular conventions for the sectioning of words for list processing are more or less dictated by the system design of the computer being used. Having established the convention for sectioning a standard list element, the requirements of list processing will be satisfied by merely extending the set of logical and arithmetic operations to the individual subfields of the list element. A reasonable set of such instructions for the 7090 suited to the list conventions of GHG is displayed in the Appendix.

List-Search Operations

The operation of *list searching* is the list-processing equivalent of the table look-up (or *convert*) operation that is common in standard addressable memory computers. The list-search mode *is* in fact a table look-up process where the data are stored in lists instead of in tabular arrays. Such processes occur far more frequently in the context of list processing than might be expected because, as we have already indicated, the ability to compute the address of a desired datum is denied the list programmer. Hardware implementation of the list-search operations could therefore be expected to return significantly greater dividends in computation efficiency than the corresponding table look-up instructions.

The list-search instructions would allow one to examine rapidly (at the rate of one entry per machine cycle) every list element of a given list for the appearance of a specified quantity in a specified field, or for the existence of a given bit pattern within the list element. The circuit logic for such instructions will be quite similar to table look-up logic, except that the address of the next word examined in sequence is extracted from the link field of the list element rather than computed from the previous address. The process terminates with the retrieval of the list element containing the desired information, or when the link field yields the list termination symbol (a zero address in GHG).

Because of the possibility that a programming or machine error might yield a re-entrant list (a "circular" list), all list-search processes must be monitored to be certain that they eventually terminate. This could be accomplished most conveniently by counting the elapsed machine cycles during the search process. If the count should run beyond a predetermined maximum (for example, an integer equal to the total number of words of core storage, and hence, the maximum length of a linear list), the process would terminate, and a "list check" condition would be set.

Bulk Storage of Lists

Any truly practical system for processing list-organized information must clearly provide for the efficient transfer of lists to and from intermediate bulk storage. Unfortunately, the transfer of a list in either direction between cores and tape poses several serious problems.

In general, a list will contain among its data entries the names of a number of other lists. For obvious reasons, the machine name of a list is the address of its initial element in storage. When a list is transferred from cores to tape, the registers released by the transfer are, of course, returned to the list of available storage for reuse. Upon being returned to core storage, that list will almost certainly occupy a completely disjoint set of registers. Consequently, a list that suffers the round trip from cores to tape and back will generally suffer a corresponding change of name.

The problem of list transfer can be solved in a number of different ways. Whatever alternative is chosen, however, the process of list input and output requires considerably more computation than is the case with ordinary block transfer of storage. If list input and output is to be performed simultaneously with computation, as indeed it must, the buffer computer must be capable of performing almost all of the fixed-point logic available to the central processing unit, with the obvious exception of multiplication and division. The availability of such a buffer computer would yield a rich harvest of additional benefits, not the least of which would be the ability to return spent lists to available storage while the main computer simultaneously pursues the main program.

Conclusion

The introduction of generalized indirect addressing alone would radically improve the adaptability of standard computer logic to the requirements of list manipulation. By further modifying the computer logic as indicated above so that all superfluous operations are eliminated in performing the primitive list processes

[2] Because of the paucity of available indicator bits in the 7090 instruction format, it would undoubtedly be necessary to pre-empt the instruction tag bits to specify a generalized indirect-address chain. Special instructions would then be required to permit long indirect-address chains to be assembled by a sequence of instructions without explicitly addressing the index register assigned to EAR. This point is treated in detail in the Appendix.

enumerated in GHG, equally substantial gains may be achieved in matching the computer to the problem. But no system will be suitable for wholesale list processing without the addition of an input-output buffer computer.

Because fifteen bits of each list element must always be assigned the relatively unproductive "housekeeping" duty of indicating the next list element, the thirty-six bit fixed word length of the 7090 is considerably below optimum size for list organization. If one stores a fifteen-bit list name in an element, a wholly inadequate number of bits remain for classification of the appended list, while floating point numbers of useful precision can be stored only indirectly (the *d*-words of GHG). The limited word size makes it most difficult to set up bi-directional lists, where each element points to its predecessor as well as to its successor. Such lists, while rarely necessary, are extremely useful upon occasion.

Appendix

A. List-Processing 7090

The suitability of the subject computer system for list processing may be vastly increased by moderate modification of the hardware logic. Unfortunately, the inadequate instruction word length makes it difficult to design an augmented system that is completely compatible with the standard 7090 instruction logic. We wish to recall, too, that this Appendix is included merely as an illustration; engineering and economic feasibility for such modification has not yet been determined.

Because the variable-length multiply (VLM) instruction count field can encroach upon one or the other (but not both) of the indirect-address flag bits, indirect addressing was made contingent upon the conjunction of two bits. In our augmented system, we require that the VLM instructions be filtered out before interpretation of the indirect-address field, in order that a single bit may be assigned the latter task. The remaining bit may then be used to signal the loading of EAR with the effective address of the flagged instruction.

A typical 7090 instruction, then, that is indirectly addressable in the standard system, is to be indirectly addressable in the general sense. Bit 12, however, is now used to signal the loading of EAR, while bit 13 is retained as the indirect address bit (Fig. 1). Bits 14-16 are now interpreted as the *mode* of indirect addressing. When set to zero, the instruction is interpreted in the standard way, with bits 18–20 interpreted as an index register tag. When the mode of an indirectly addressed instruction is not zero (Fig. 2), bits 17–20 are interpreted as a *key* to specify the desired address chain. Clearly, such an instruction is not indexable. However, the fact that list-processing operations rarely involve the traditional use of index registers mitigates this loss to a great extent. When used as the EAR, the contents of an index register may be extracted with the instructions described below.

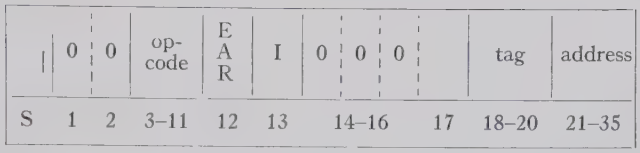


Fig. 1—Modified instruction format with mode zero.

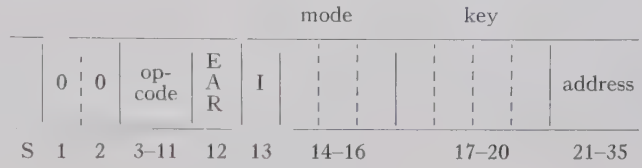


Fig. 2—Instruction indirectly addressed in the general sense (mode>0).

Table I specifies in detail a possible interpretation of the generalized indirect-addressing code. In each case, it is assumed that bit 13 is "on."

TABLE I

Mode	Key
$M=0$	Bits 18–20 are interpreted as the tag of an ordinary 7090 indirectly-addressed instruction.
$M=1, 2, 3, 4$	M is interpreted as the length of the indirect-address chain. The first M bits of the key are interpreted as a pattern specifying whether the nth replacement is to be from the decrement (1 bit in pattern) or the address field (0 bit in pattern).
$M=5$	Key is interpreted as a 4-bit binary number specifying the length of an indirect-address chain where all replacements are from the address.
$M=6$	Key is interpreted as a 4-bit binary number specifying the length of an indirect-address chain where all replacements are from the decrement.

As an example, the indirectly addressed instruction CLA, Y with mode 3 and key 110 would place in the accumulator the word at the address obtained by replacing the number Y with the decrement field of Y, say Y_1, replacing Y_1 with the contents Y_2 of the decrement field of Y_1, and replacing Y_2 with Y_3, the contents of the address field of Y_2. Each replacement requires the insertion of one extra E cycle in the execution of the instruction, during which time the contents of the address register of the CPU is replaced by the decrement or address field of the word at that address.

Unfortunately, the scheme above contains a certain amount of unavoidable redundancy. Thus, an indirectly-addressed instruction with mode 5, key 1, is equivalent to the same instruction with mode 1, key 0. Similarly, an instruction with mode 4, key 1111 is equivalent to one with mode 6, key 4, and so on. It is assumed, of course, that all normal indexing is suppressed for the case of mode>0.

The 7090 Effective-Address Register

Our augmented system will make use of an ordinary

TABLE II

EXTENSIVE SET OF LIST-PROCESSING INSTRUCTIONS*

Clear Add Address to Accumulator, *Y* Clear Add Decrement to Acc, *Y* Clear Add Spfx to Acc, *Y* Clear Add Pfx to Acc, *Y* Clear Add Tag to Acc, *Y*	The accumulator is cleared and the address (decrement, signed prefix, prefix, tag) field of the word at *Y* is placed in the low-order 15 bits of the accumulator.
Add Address to Acc, *Y* Sub Address to Acc, *Y* And Address to Acc, *Y* Or Address to Acc, *Y* Add Decrement to Acc, *Y* . . . Add Spfx to Acc, *Y* . . . Add Pfx to Acc, *Y* . . . Add Tag to Acc, *Y* . . .	The designated operation is performed with the designated field of *Y* and the accumulator. The low-order bits of the field specified are paired with the low-order bits of the accumulator, which is treated as a long register. The result is in the accumulator.
Store Acc in Address, *Y* Store Acc in Decrement, *Y* Store Acc in Spfx, *Y* Store Acc in Pfx, *Y* Store Acc in Tag, *Y*	The required number of bits from the low-order end of the accumulator are stored in the designated field of location *Y* (15 for address, dec; 3 for spfx, tag; 2 for pfx). Excess bits in the accumulator are ignored.
Replace Address by Address, *Y* Replace Dec by Address, *Y* Replace Add by Dec, *Y* Replace Dec by Dec, *Y* Replace Tag by Tag, *Y* Replace Tag by Spfx, *Y* Replace Spfx by Tag, *Y* Replace Spfx by Spfx, *Y* Replace Pfx by Pfx, *Y*	The first-specified field of the accumulator is replaced by the second-specified field of the word at location *Y*.
Store Address in Address, *Y* Store Dec in Add, *Y* Store Add in Dec, *Y* Store Dec in Dec, *Y* Store Tag in Tag, *Y* Store Tag in Spfx, *Y* Store Spfx in Tag, *Y* Store Spfx in Spfx, *Y* Store Pfx in Pfx, *Y*	The first-specified field of the accumulator is stored in the second-specified field of the word at location *Y*.
Add Address to Address, *Y* Add Address to Dec, *Y* . . . Sub Address from Address, *Y* . . .	The first-specified field of the word at location *Y* is added (subtracted) to the second-specified field of the accumulator. Overflow may occur.
And Address to Address Acc, *Y* And Address to Dec Acc, *Y* And Dec to Add Acc, *Y* And Dec to Dec Acc, *Y* And Tag to Tag Acc, *Y* And Tag to Spfx Acc, *Y* And Spfx to Tag Acc, *Y* And Spfx to Spfx Acc, *Y* And Pfx to Pfx, *Y* Or Address to Address Acc, *Y* . . .	The designated operation is performed between the designated fields of the accumulator and the word at location *Y*, the result appearing in the accumulator.
And Address to Address Storage, *Y* . . . Or Address to Address Storage, *Y* . . .	The designated operation is performed between the designated fields of the accumulator and the word at location *Y*, the result appearing in storage.
Search for Decrement, *Y* Search for Address, *Y* Search for Sign, *Y* Search for Spfx, *Y* Search for Pfx, *Y* Search for Tag, *Y*	The designated field of the word at location *Y* is compared with the corresponding field of the accumulator. If they are identical, the address field of the accumulator is replaced by the address *Y*. If not, *Y* is replaced by the decrement field of the word at location *Y*. If that quantity is zero, the operation terminates and the computer skips the next instruction. If not, the operation is repeated with the new value of *Y*. A count is kept of the number of replacements of *Y*. When that number exceeds *N*, the number of words of core storage for the computer, a list-check light goes on, and the computer halts.
Search for Bit Pattern, *Y*	The word at location *Y* is examined for a *one* bit in each bit position indicated by the pattern in the indicator register. If the condition is satisfied, the address field of the accumulator is replaced by the address *Y*. If not, *Y* is replaced by decrement field of the word at location *Y*. If that quantity is zero, the operation terminates and the computer skips the next instruction. If not, the operation is repeated with the new value of *Y*. A count is kept of the number of replacements of *Y*. When the number exceeds *N*, the number of words of core storage for the computer, a list-check light goes on, and the computer halts.
Compare Address with Address, *Y* Compare Address with Dec, *Y* Compare Dec with Add, *Y* Compare Dec with Dec, *Y* Compare Tag with Tag, *Y* Compare Tag with Spfx, *Y* Compare Spfx with Tag, *Y* Compare Spfx with Spfx, *Y* Compare Pfx with Pfx, *Y*	The first-designated field of the accumulator is compared with the second-designated field of the word at location *Y*. If they are identical the computer takes the next instruction in sequence. If not, the next instruction is skipped.
Compare Accumulator with Address, *Y* Compare Acc with Dec, *Y* Compare Acc with Tag, *Y* Compare Acc with Spfx, *Y* Compare Acc with Pfx, *Y*	The low-order end of the accumulator is compared with the designated field of the word at location *Y*. If they are identical, the computer takes the next instruction in sequence. If not, the next instruction is skipped.
Transfer on Zero Address, *Y* Transfer on Zero Decrement, *Y* Transfer on Zero Tag, *Y* Transfer on Zero Spfx, *Y* Transfer on Zero Pfx, *Y* Transfer on Nonzero Address, *Y* Transfer on Nonzero Decrement, *Y* Transfer on Nonzero Tag, *Y* Transfer on Nonzero Spfx, *Y* Transfer on Nonzero Pfx, *Y*	The instruction at location *Y* is executed next if the specified field of the accumulator is zero (nonzero).

* All instructions are to be indirectly addressable in the generalized sense.

index register to serve the purpose of EAR. An instruction *Select EAR* will select the index register designated by the tag of that instruction as the particular register to be addressed by all EAR flagged instructions. The instance of an EAR bit in a given instruction will cause the last computed effective address for that instruction to replace the previous contents of EAR. The number in EAR is retrieved by tagging an instruction in the ordinary way with the register specified as EAR.

One additional instruction makes it possible to link together short single-instruction indirect-address chains into one long chain. The instruction *Set Address, Y* is in fact a no-op instruction where the effective address is computed, stored in EAR, and the *EAR Trigger* turned on. Any instruction that follows, sensing the *EAR Trigger* on, will have its address field replaced by the contents of EAR, at the same time turning the *EAR Trigger* off. The following sequence of instructions will set up the indirect-address chain specified by the bit pattern *10110001010*:

Set Address, *Y*, mode 4, key 1011
Set Address, 0, mode 4, key 0001
Next Instruction, 0, mode 3, key 010.

Field Logic and List-Search Operations

Table II is a rather extravagant listing of new instructions that are especially suited to list-processing operations. The listing is extravagant in the redundancy it contains (some of the commands perform operations identical with existing 7090 instructions, some are identical with others in the table), and in the lavish introduction of new commands with insufficient experience to testify for their presumed usefulness. It is expected that research in the simulation of a list-processing computer will result in drastic pruning of the listing. Furthermore, by accepting certain list structure conventions at the outset, one may dispose of additional groups of commands. A final justification for our prodigality is the fact that, upon occasion, the listing of the same instruction under different names affords somewhat greater clarity of exposition.

BIBLIOGRAPHY

[1] J. Bosset, "Certain aspects of the logical design of the Gamma 60," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France, 1960. (In French.)
[2] H. Gelernter, J. R. Hansen, and C. L. Gerberich, "A FORTRAN-compiled list processing language," *J. Assoc. Comp. Mach.*, vol. 7, pp. 87–101; April, 1960.
[3] H. Gelernter, "Realization of a geometry theorem-proving machine," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France; 1960.
[4] J. McCarthy, "Recursive functions of symbolic expressions and their computation by machine," *Commun. Assoc. Comp. Mach.*, vol. 3, pp. 184–195; April, 1960.
[5] A. Newell and J. C. Shaw, "Programming the logic theory machine," *Proc. Western Joint Computer Conf.*, Los Angeles, Calif., February 26–28, 1957; pp. 230–240.
[6] J. C. Shaw, A. Newell, H. A. Simon, and T. O. Ellis, "A command structure for complex information processing," *Proc. Western Joint Computer Conf.*, Los Angeles, Calif., May 6–8, 1958; pp. 119–128.
[7] V. H. Yngve, "The COMIT system for mechanical translation," *Proc. Internatl. Conf. on Information Processing*, UNESCO, Paris, France; 1960.

The Simulation of Three Machines Which Read Rows of Handwritten Arabic Numbers*

L. A. KAMENTSKY†, MEMBER, IRE

Summary—Three machines have been simulated using an optical scanner and the IBM 704 computer. Each of these simulated machines has read documents containing *rows* of handwritten Arabic numbers. Sample numbers were produced by at least 20 people for each simulation study. The three machines simulated differ in the control required of the writer during document preparation and in the complexity of the machines. Writing controls were required for the preparation of the first two types of documents. A section of this paper concerns experiments with and a mathematical model of controlled writing. The third simulated machine was applied both to numbers written within preprinted boxes and to numbers written without any guide marks. About one per cent of 2180 of these numbers were misread as the wrong character. This error rate is based on a sequential experiment in which the recognition logic is constructed from all characters not recognized and thus rejected prior to each input character. Numbers, when rejected, cause the program to identify them from a table. Their structure is then entered into the recognition logic. The rejection rate decreased throughout the experiment. The last rejection rate was about 10 per cent.

* Received by the PGEC, July 29, 1960.
† IBM Res. Ctr., Yorktown Heights, N. Y. Formerly with Bell Telephone Labs., Inc., Murray Hill, N. J.

I. Introduction

IT has often been stated that the input is the weakest link of the functional parts of an integrated data-processing system. Such languages as punched cards, paper tape, or magnetic tape for inputs to a computer are only intermediate in many cases. These languages result from the inability of present data processors to understand the language understood by people. This language translation from documents produced by humans to cards or tapes may account for a major part of the cost of equipment and salaries in a machine data-processing system. For reasons similar to this, in the last 30 years there have been investigations of machines for reading printing and handwriting.

To date, some machines have been developed for the reading of print of a fixed-type font. At present, no reliable machine exists for reading print of varied styles. Although there are some preliminary studies in the literature,[1,2] certainly no machine can read handwriting as yet.

This is a paper on the simulation of three character reading machines. An optical scanner[3] and the IBM 704 computer are used to perform this simulation. Each of these machines can read documents containing rows of handwritten Arabic numbers. These three machines differ in:

1) The control required of the persons producing these documents.
2) The complexity of the reading systems themselves.

For each method, an IBM 704 simulation program was written. Documents written by at least 20 people were used in the simulation process.

The three machines simulated will be referred to in the following discussion as method 1, method 2, and method 3. A sample of the documents used for method 1[4] is shown in Fig. 1. A document consists, in each case, of a row of numbers written with respect to visible indicia. In method 1, the writer was told to write around two dots and between two guide lines. In method 2, the writer was told to write using four lines as a guide. Sample documents for method 2 are shown in Fig. 2. In method 3, one half of the documents scanned were produced by asking the writer to write in boxes. The other half were produced by asking the writer to write 4 or 5 numbers in a row. Sample documents for method 3 are shown in Figs. 3 and 4. In all cases, the writer was told to make "open top" "fours." It is important to note that the writer was told to write using constraints.

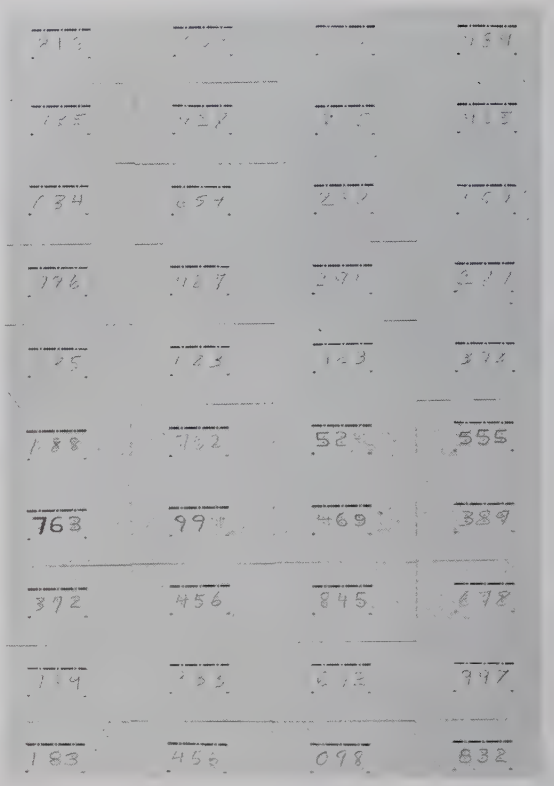


Fig. 1 –Some sample documents scanned in the study of method 1.

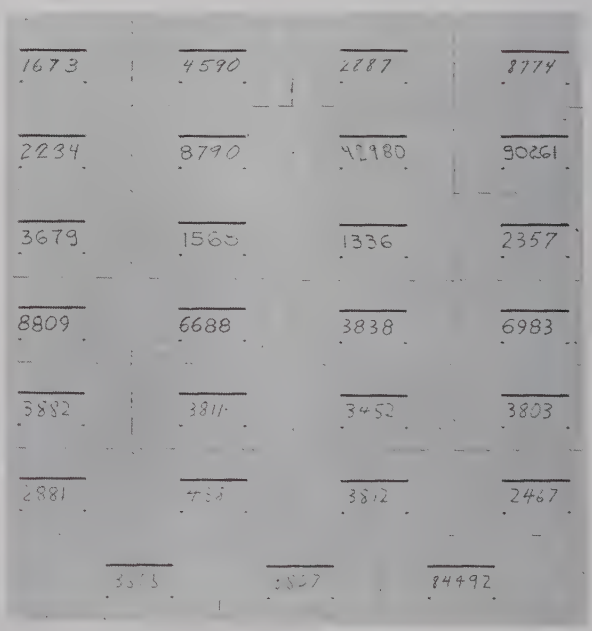


Fig. 2—Some sample documents scanned in the study of method 2.

In many cases, he did not follow them. These badly-written documents were included among the better ones and were scanned and read with some success.

In method 1, the writer is constrained by the dots to write within certain locational, rotational, and size bounds. By using lines in method 2 instead of dots, the bounds on horizontal location, tilt, and size are extended as compared to method 1. In method 3, both the horizontal and vertical bounds are extended with respect to

[1] S. H. Unger, "Pattern detection and recognition," Proc. IRE, vol. 47, pp. 1737–1752; October, 1959.
[2] J. S. Bomba, "Alpha-numeric character recognition using local operations," *Proc. EJCC*, Boston, Mass., December 1–3, 1959; pp. 218–224.
[3] W. H. Highleyman and L. A. Kamentsky, "A generalized scanner for pattern and character recognition studies," *Proc. WJCC*, San Francisco, Calif., March 3–5, 1959; pp. 291–294.
[4] Method 1 has been described by T. L. Dimond, "Devices for reading handwritten characters," *Proc. EJCC*, Washington, D. C., December 9–13, 1957; pp. 232–237.

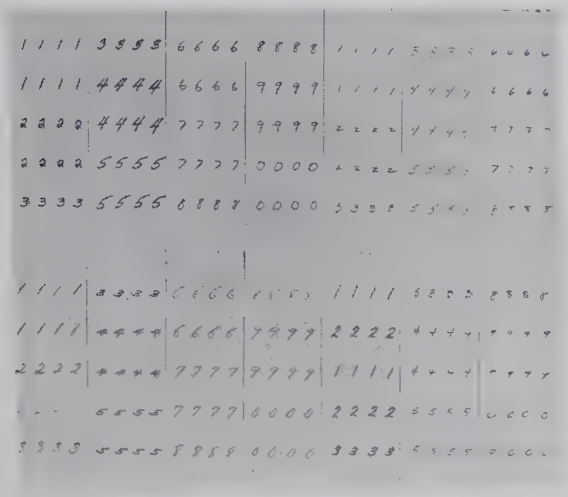


Fig. 3—Some sample documents scanned in the study of method 3 (with boxes).

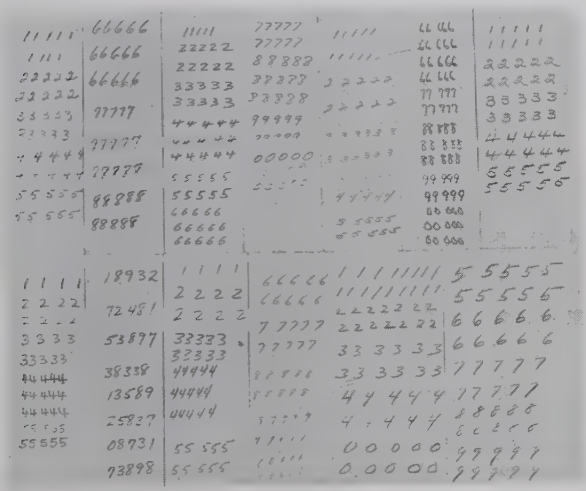


Fig. 4—Some sample documents scanned in the study of method 3 (no constraints).

method 1. It is obvious that the use of controlled or constrained writing may hamper the writer in at least two ways: 1) he might not write as fast as he could have, had he not been using a controlled writing system; 2) the writer may not be able to adhere to the control. Thus some errors may be introduced by a reading machine structured to depend on the control. However, balanced against this difficulty of control is the requirement for a less complex reading machine. Some experiments with writing tests are described in the Appendix. The results of these tests are compared to a current theory of the information capacity of the human motor system.

Fig. 5 shows a flow chart describing the general simulation process. Each method of simulation may be broken down into three steps corresponding to sections of this chart.

In the first step, a program is run on the IBM 704 computer to generate a program tape to be used in directing the operation of a scanner. The optical scanner[3] which was used for this study uses a magnetic

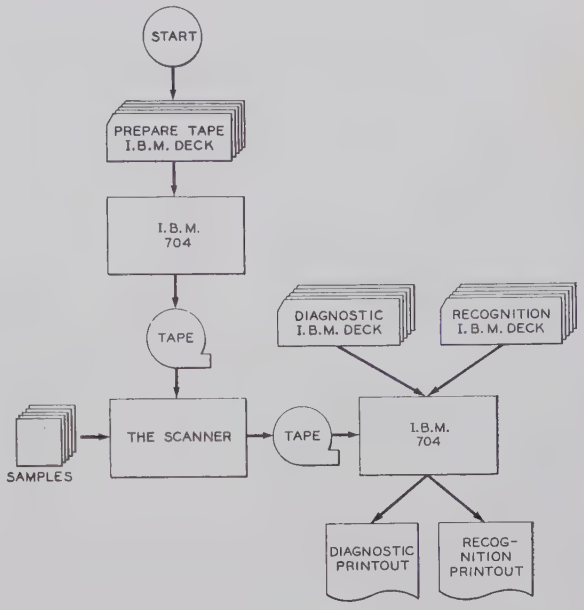


Fig. 5—Flow chart of this study.

tape to specify the coordinates of a line-by-line scan on the face of a cathode-ray tube. The scan pattern to be generated can be arbitrarily chosen by the experimenter.

In the second step, documents are scanned in a way dictated by this tape. The light reflected from the document for each segment of a scan is used to determine whether the segment did nor did not cross a mark on the document. A sequence of bits giving this information is recorded on a second magnetic tape at the scanner output.

In the last step, this output tape is processed by the IBM 704 computer. Programs were written to diagnose the performance of the scanner and the recognition procedures and to yield diagnostic and recognition information on the various numbers.

This paper will begin with a discussion of "mapping scans" and how they are used in all three methods. Each of the three methods of recognition will be described, and the experimental results achieved with each will be presented.

II. General Description

Machine pattern or character recognition can be considered to be a two-step process. To recognize n patterns a machine must:

1) Evaluate m parameters present in the input which describe the n patterns in some way. The m parameters must be extractable by a realizable and, if possible, economical machine.
2) Find decision criteria which completely, uniquely, and reliably define all n patterns using the m parameters.

In the machine studied here, the number of parameters to be decoded at one time is small (7 to 15). Thus, relatively simple decoding logics can be used to enumerate a

large percentage of the most probable states of the parameters.

In this study, the particular parameters extracted from the documents are invariant, over a wide range, to many variations in the written numbers. The numbers are either constrained with respect to, or the parameters used are invariant to:

1) Changes in form which do not alter the meaning to a human reader.
2) Line width and density.
3) Position, tilt, and size.

The parameters, however, are sensitive to variations which do alter the meaning to the human reader.

The parameters used are here called the "loop configurations" of numbers.[5] In the "loop space," Arabic numbers are considered to be made up of lines forming fully-closed loops or loops which are open on one, two, or three sides. A "one" in this picture is a line loop (or two loops) open on three sides. A "two" can be described as a loop open to the left, and a "nine" is a closed loop above a loop open to the left and possibly to the bottom. A subset of the possible ways of forming two loops which are fixed above each other and which can be open in four given directions is adequate to describe many variations of the 10 numbers.

It is possible to computer-process a matrix-element representation of each scanned number so as to generate codes describing the loop configurations.[5] However, in these studies a different scanning procedure was used to map the Arabic numbers onto an ordered array of bits. The document was scanned along many carefully arranged straight-line segments and dots. For each segment or dot, a (digital) record was made (on magnetic tape) as to whether or not that line segment or dot covered any black areas. The magnetic tape containing these records for all the segments was the output of the scanning step. This tape was then processed by the IBM 704 computer to carry through the character recognition algorithm. The scan patterns described in this paper were chosen after many other possible scan patterns had been tried and discarded.

For clarity, let us describe in detail the mapping scan used in method 3 and how the bits derived from this scan were processed to assemble codes representing the loop configurations of the numbers. Consider the pattern of four lines and a dot shown on the left in Fig. 6. The scan traversed such a pattern centered at 5000 different places, arranged in a square grid of 50 rows and 100 columns. The document scanned contained a row of 4 or 5 Arabic numbers, as shown in Fig. 6. For each point of the grid, the computer assembled (from the scanner output tape) a code number from 0 to 7 according to the following formula:

$$C_1 = \begin{cases} 1 & \text{if all of the left-hand segments cross black} \\ 0 & \text{if not} \end{cases}$$

$$C_2 = \begin{cases} 2 & \text{if the right-hand segment crosses black} \\ 0 & \text{if not} \end{cases}$$

$$C_4 = \begin{cases} 4 & \text{if the dot crosses black} \\ 0 & \text{if not} \end{cases}$$

$$\text{Code number} = C_1 + C_2 + C_4.$$

Fig. 6—The mapping scan for method 3 and a typical document.

This code is called the loop-configuration code. A typical map (of the codes arranged in the grid) is shown in Fig. 7. For clarity, the code zero has been suppressed on this map and codes ≥ 4 (which represent the character body) have been replaced by an asterisk. Also, lines have been drawn around the character bodies. The loop configurations along any column of a given written numeral can be deduced by looking down that column of loop-configuration codes on the map.

Scans such as that described above were generated (indirectly) by the IBM 704 using a program to write the scanner input tape. An interpretive program was written so that the scan could be changed for different experiments by simply punching a few cards. The complex processes which generated the mapping scan tape were done only once for each experiment and applied to all documents for which that scan pattern was tried. The magnetic tape output of the scanner contains groups of bits ordered to allow a simple assembly of the loop-configuration codes, at each grid point. These groups of bits are further ordered with respect to successive grid point rows and columns. One tape file specifies the scans of one document.

In processing the scanner output tape for the three simulation studies, the computer first assembled the loop-configuration codes. In method 3, the two-dimensional array of codes was reduced to another kind of code by a process which will be described later, and this code was compared with code lists for each of the 10 Arabic numerals. In method 2, not only was a loop-configuration code (different from the one for method 3) formed by the scanner, but also a vertical-position code was formed by which the computer is later able to discover which grid point of a vertical column is vertically centered. The computer program searched each vertical column of codes to find the proper one, and only these vertically centered codes were thereafter used. In method 1 the scanner formed, besides a loop-configuration code, a combined vertical-horizontal position code. The computer searched for the codes which were cen-

[5] The use of loop descriptions is not novel: see, for example, N. Rochester, *et al.*, U. S. Patent No. 2,889,535; June 2, 1959.

Fig. 7—Method 3 code map.

tered on the numbers, and thereafter used only the centered codes. Each of the three recognition methods will now be discussed in more detail.

III. Recognition of Numbers Constrained by Two Dots (Method 1)

A. The Scan Pattern for Method 1

Fig. 8 shows the pattern of scan used at each grid point with the segment weights. Fig. 9 shows the scan superimposed on the 2-dot type of document. This scan is centered at 1100 grid points arranged in 11 rows and 100 columns. The proportions are shown in Fig. 9 as they were in the actual scan. The numbers of Fig. 8 refer to the weights to be added in forming the codes if the indicated segment or segments cross a mark. This scan consists of 16 segments. Intersections of marks with the upper 5 segments indicate the position of the scan center with respect to the upper registration bar. This bar is fixed in position with respect to the dots. The code representing these 5 segments is called the position code and is separate from the code for the other 11 segments; the recognition logic finds the properly registered scan by searching for the position code "23."

There are 11 scan segments used in forming the loop-configuration code. The code for the properly registered set of 11 segments is decoded by the recognition logic. Note that the scans with weights 1, 8, and 16 are multiple. Written pencil lines are continuous lines, and the accuracy of recognition can be increased by taking advantage of this fact. In this system, a weight of 1 occurs only when both segments marked 1 in Fig. 8 cross a mark. Similarly, the 3-scan segments of weight 8 and the 2-scan segments of weight 16 are "anded." This "anding" over sector areas is, in effect, a search for openings in each of these directions by detecting the presence of no mark in each area. This type of scanning is effective for finding such things as openings in "fives" to distinguish them from "sixes," or for integrating out tails on "threes" or "sevens."

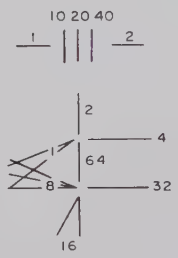


Fig. 8—The mapping scan for method 1.

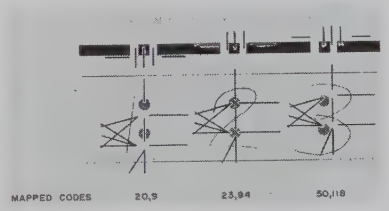


Fig. 9—The scan for method 1 superimposed on a typical document.

Two other points must be stated here. The guide lines and dots were printed with a blue ink invisible to the scanner. Thus, the scanner could see only the numbers and the registration bar, and only these are reflected in the codes it formed. Also, the vertical excursion of the scan centers did not normally allow the registration segments (the upper five lines of the pattern) to intersect any written line.

B. The Recognition Logic for Method 1

The simplest recognition logic would compare the registered code against the 128 possible 7-bit codes and generate the proper recognition by table lookup. If the numbers were always written to the required constraint, this simple system would work. Since the number of samples obtained with improperly written numbers is considerable, some error correction was built into the program.

At least two schemes are available for error correction. In the first, the loop-configuration codes at nearby grid points, rather than the one code at the registered point, may be operated upon by the logic. The logic would be of a form such as to assign the recognition code to the majority code in a given group of codes in the vicinity of the registration position. Only the very simplest of this class of correction methods was used. The loop-configuration codes from the grid points above and below this registered grid point were compared to the registered loop-configuration code. If these two codes were alike and differed from the registered code, the registered code was replaced by this code.

The second technique is based on the fact that of the 128 possible codes, only about 20 belong to legitimate configurations of the 10 Arabic numerals. This fact was used by comparing the registered code to the 20 legitimate codes. If the registered code was not a legitimate code, the registered code was replaced by the code at one of the eight neighboring grid points. (There are eight adjacent scan centers bordering on almost all scan centers.) This new code was tested for legality. If the code was not legitimate, it was replaced by another of the eight neighboring codes. The process was repeated until either a legitimate code was found or all eight neighboring codes were tried. If the process failed, the number was rejected.

C. The Results for Method 1

Since the primary interest of these studies was to design a system to read unconstrained writing, the number of samples used for methods 1 and 2 was only large enough to show feasibility. Two types of samples were scanned using method 1. In the first sample, 61 cards, each with 3 numbers, were written by 20 people. Two cards were incorrectly read. The 3 numbers misread on these were all malformations (the only miswritten numbers that could not be corrected). The errors, shown in Fig. 10, are the "two" on the first document (called a "three"), and the "one" (called a "seven") and the "seven" (called a "three") on the second document.

A second set of samples consisted of a number of cards with markings on them. The purpose of these cards was to test variations in registration and the performance of the scanner and logic with respect to size, line density, and smudging. Some of these samples are shown in Fig. 11. These were all read correctly including the "888" on the third sample and the "000" on the fourth sample.

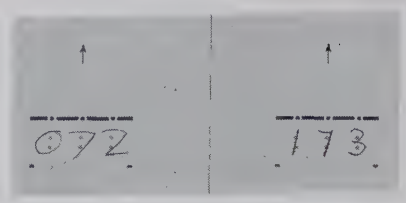


Fig. 10—All cards read incorrectly with method 1.

Fig. 11—Some test sample documents scanned in the study of method 1.

IV. 4-Line Constrained Writing (Method 2)

A. The Scan Pattern for Method 2

The scan pattern at each grid point for method 2 is shown in Fig. 12. This scan is centered at 1100 grid points arranged in 11 rows and 100 columns.

The pattern consists of 17 segments; 2 of these are used for vertical registration and 15 are used for the recognition logic. For each of the 1100 grid points, bits representing the mark crossings of scan segments 1 and 2 are "anded," the "and" condition indicating that the remaining scan segments are properly positioned along the two inner guide lines.

To allow for wide variations in the tilt of the 10 numbers, the code representation using the lower 15 scans was broken into two parts. One code was assigned to the top part of the number (called the top code); the second code was assigned to the bottom part of the number (called the bottom code). This allowed the system to recognize the top and bottom of each number independently and to put them together in the final logic. This results in a simpler and more reliable logic for wider variations in number formation.

The scan weights are tabulated below. The numbers of the segments correspond to the segment numbers of

Category	Scan Segments "Anded"	Weight
Registration	1, 2	(Registered)
Top Code	3, 4, 5	1
	6, 7	2
	8	4
	15, 16	8
Bottom Code	9, 10, 11	1
	16, 17	2
	14	4
	12, 13	8

Fig. 12. The pairs of numbers shown in Fig. 13 represent the top and bottom codes for the indicated positions of the pattern.

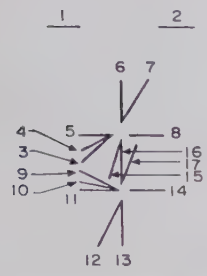


Fig. 12—The mapping scan for method 2.

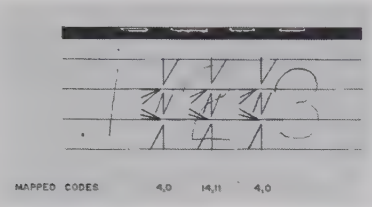


Fig. 13—The scan for method 2 superimposed on a typical document.

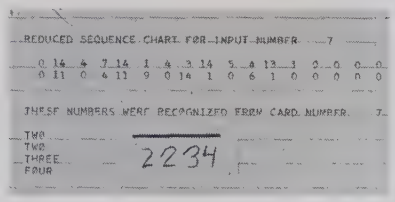


Fig. 14—A recognition printout for method 2.

B. The Recognition Logic for Method 2

The input to the recognition logic for method 2 is an ordered sequence of 100 pairs of vertically registered 4-bit codes. A successful recognition logic must do two things: it must separate the numbers on the documents, and it must recognize them. The logic examines the sequence of code pairs, comparing it against a table of allowed and disallowed code sequences for each Arabic numeral.[6] Included in each list are end-of-character codes indicating that the current number should be entered in the recognition list and the logic reset to read the next number.

The first step in the processing of the scanner output tape is to determine the sequence of 100 registered code pairs per document. (A document contains 4 or 5 written numbers.) Next, these codes are processed to reduce the size of the list of code pairs. The reduction criterion used is this: five contiguous codes of each of the two strings of 100 codes are tested at one time; this test may be likened to a window which covers the first five codes and which shifts to the right one code place after each test. For each test, if three out of a group of five contiguous codes (in the window) are alike, the test condition is met and these codes are noted. The code lists derived as described above are further reduced by eliminating all but one code of any group of contiguous identical codes. A typical document, a sequence of reduced code pairs, and a recognition list are shown in Fig. 14.

Next, the reduced code lists are examined by a table lookup logic. This logic consists of lists of allowed and forbidden codes which are compared in sequence with the pairs of reduced codes. The results of these comparisons define certain features of each input numeral. The top codes are processed first and compared in sequence with eight lists of codes to define one of eight features. A code at the end of the feature must also be found to separate the numbers. Such an end-of-feature code sequence is found for every feature found. Typical features are a full loop, a loop open to the right, a single line, etc. A typical end-of-feature code sequence is: marks to the left only, marks on both sides, marks to the right only. When an end-of-feature sequence is found, the feature defined is recorded and the process is repeated for the bottom sequence of codes beginning from the end of the last feature. For example, the reduced code sequences on the top of 4, 14, 5, 4 (a form of the feature, loop open to the left), and on the bottom of 4, 11, 5, 4, (a form of the feature, loop open to the right), would be sufficient to identify the two features which define the number "two" and also to reset the logic.

C. The Results from Method 2

The logic for method 2 was designed using the same samples as were used for method 1. The scanner tape produced by the samples in method 1 was used directly with the program of method 2 by simply ignoring the horizontal registration bits. An accuracy of 100 per cent was obtained for these samples. Thirty more documents using the constraint of method 2 were obtained from 10 people and were used with the first sample in redesigning the code tables and for determining a good scan pattern for method 2. Using the logic designed with these two sets of samples, 129 new numbers, some of which are shown in Fig. 2, were used to test this logic. There were four misrecognitions, and there were no errors in separating characters with these samples. The misrecognitions resulted in three cases from unresolved line segments, and in one case from a "six" written to look almost like an "eight."

V. Recognition of Unconstrained Numbers (Method 3)

A. The Scan Pattern for Method 3

The 5-segment pattern for method 3 is shown in Fig. 6. This scan is centered on 5000 grid points on the written document, forming an array of 50 rows and 100 columns on the writing area. The relative size of this scan with respect to the document is shown in the example in Fig. 6. The generation of loop-configuration codes for method 3 has already been discussed in Section II.

[6] This is similar to a decision method of E. C. Greanias, *et al.*, "Design of logic for recognition of printed characters by simulation," *IBM J. Res. & Dev.*, vol. 1, pp. 8–18; January, 1957.

B. The Logic for Method 3

The logic for method 3 must deal with a two-dimensional array of codes like that shown in Fig. 7. This logic consists of two subprocesses: one of these processes reduces the two-dimensional array to a one-dimensional sequence of codes in a program called the "line-feature generation program;" the second process reduces this sequence of codes to a single code (as was done in method 2) in a second program.

The line-feature generation program will be discussed first. Its objective is to generate two codes which correspond to the top and bottom codes of method 2 for each column of the 50 scans of method 3. It does this by comparing the sequence of loop-configuration codes (from the top to the bottom of each column) with required sequences of codes for each of 10 line features and an end of character indication.

The rules used to establish line features from each column of 50 loop-configuration codes will now be described. The codes of each column are first scanned by the program using a three-code window. If any code is contiguous to two codes which are alike and differ from it, the code is replaced by the adjacent codes.

Next, one or more line features are established for each column of loop-configuration codes by comparing these codes with required sequences for each possible line feature. Column 1 of Table I lists the 10 line features and their assigned codes. Column 2 shows the character configurations for each line feature. The third column of Table I lists the required sequences for each line feature. In this column, C_i is the loop-configuration code i, where $i=1, 2, 3$, or m. $i \geq 4$ (asterisk on the Fig. 7 code map) corresponds to a mark and is denoted by the symbol m. In certain cases, the line features are further differentiated by using the ratios of the lengths of loop-configuration code sequences. Such additional conditions are listed in the fourth column of Table I. Here, the symbol N_i refers to the length of the code sequence i. $i=A$, B, and S are used to denote any nonzero code and refer to the captions used to describe the size of the shaded areas in column 2.

The last step of the method 3 line-feature generation logic is to assign one line-feature code to the top part of each column of scan patterns and one line-feature code to the bottom. The line features are first divided into top or bottom features by using criteria based on the ratios of the length of line-configuration code sequences. The rules used for this are listed in the last column of Table I. If two or more features satisfy the necessary condition for the top or bottom code on any line, the line feature is assigned to the most closed feature. For example, loops open on one side have priority over loops open on two sides.

There are 100 line feature code-pairs per document of 4 or 5 numbers. These 100 code-pairs were reduced in number using a reduction criterion similar to that in method 2.

TABLE I

METHOD 3 FEATURES AND THEIR CONFIGURATION

CODE	CONFIGURATION	REQUIRED SEQUENCE	OTHER CONDITIONS	CONDITION TO BE TOP CODE
3	SCAN POSITION; N_S, N_1	C_m, C_1, C_0	———	$N_1 \geq \frac{1}{2} N_S$
6	N_S, N_2	C_m, C_2, C_0	———	$N_2 \geq \frac{1}{2} N_S$
7 TOP ONLY	N_A, N_3, N_S, N_B	C_m, C_3 *	$N_3 > \lvert N_A - N_B \rvert$	———
9	N_1, N_S	C_0, C_1, C_m	———	$N_1 < \frac{1}{2} N_S$
11	N_A, N_1, N_S, N_B	C_m, C_1, C_m *	$N_1 \leq \lvert N_A - N_B \rvert$ OTHERWISE THIS CONFIGURATION IS CODED AS 3 ON TOP AND AS 9 ON BOTTOM	$N_A + \frac{N_1}{2} < \frac{1}{2} N_S$ †
12	N_2, N_S	C_0, C_2, C_m	———	$N_2 < \frac{1}{2} N_S$
13 TOP ONLY		C_0 OR C_1 OR C_2 — C_3, C_m *	———	———
14	N_A, N_2, N_S, N_B	C_m, C_2, C_m *	$N_2 \leq \lvert N_A - N_B \rvert$ OTHERWISE THIS CONFIGURATION IS CODED AS 6 ON TOP AND 12 ON BOTTOM	$N_A + \frac{N_2}{2} < \frac{1}{2} N_S$ †
15	N_A, N_3, N_S, N_B	C_m, C_3, C_m *	$N_3 \leq \lvert N_A - N_B \rvert$ OTHERWISE THIS CONFIGURATION IS CODED AS 7 ON TOP AND 13 ON BOTTOM	$N_A + \frac{N_3}{2} < \frac{1}{2} N_S$ †
10 LINE CODE	N_m	C_0, C_m, C_0	NO OTHER CODES OCCUR AND $N_m \geq 6$	
2 END OF CHARACTER		C_0 (ONLY) OR C_m, C_1, C_2, C_0 OR C_m, C_1, C_3, C_2, C_0	INVALID IF C_3 OCCURS BEFORE C_m	

† If two of the codes 11, 14, or 15 occur on a column, the first code on the line is assigned to top and the other to bottom.

* The code between two C_m's or between C_0 and C_m is determined on a majority basis if two or more codes occur.

The pair of reduced code sequences at the output of the line-feature generation program could be used as the input to the recognition program of method 2. This was tried and worked reasonably well with a limited sample set. However, another decision process was studied and will be described below.

C. A Sequential List Buildup Logic

During the processing of 2180 numbers, it was found that character separation based only on the end-of-character code, discussed in the previous section, worked in all but three cases. It was further noticed that, in most cases, usually only one or two, and at most six, 4-

bit codes describing the line features were required to specify the top or the bottom of each of the separated characters. These line-feature codes were reduced to a pair of numbers and these numbers were matched against a list of code pairs (sometimes treated as two independent lists) for each character. This was done by lining up, end-to-end, in a binary register the 4-bit codes that appeared between end-of-character codes for the top of each character. This ordered sequence of bits was considered as one single octal number of one to eight digits. This was repeated for the bottom of that character. Thus, a pair of these numbers represents each of the characters on each of the documents.

The output of the line-feature generation program is a deck of IBM cards with one card for each character. Each of these cards contained the pair of octal numbers discussed above, a code number identifying the character with the original writer, and another number identifying the character itself. These cards were used as the input to a program called the list buildup program. Each card was processed in turn as shown in the flow chart of Fig. 15.

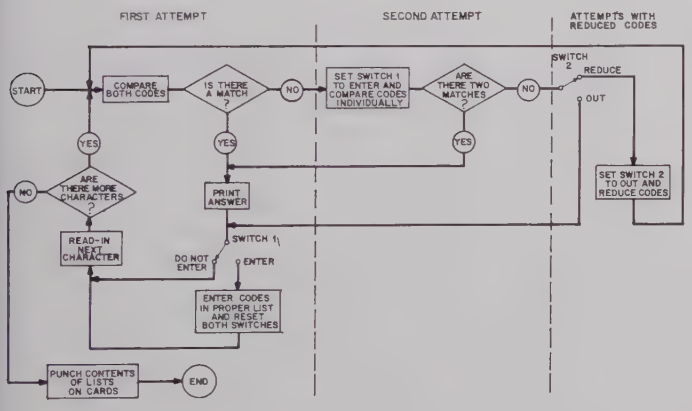


Fig. 15—Flow chart of the list buildup program.

Four distinct attempts were made to recognize each character. In the memory of the computer, there are two ordered lists of codes for each of the 10 numbers. One of these lists refers to the top code and the other to the bottom code. In the first attempt at recognition, both codes on each character card are compared with the first pair of codes on the list for the number "one." This is repeated for all list pairs of "one," then for the "two" list, etc., until both codes match a pair of codes in one of the lists. If a match is found, the number of the list is printed out and the logic is reset to read the next card which contains the next character. If there is no pair match, both of the codes on the card are entered in the proper list in memory corresponding to the correct answer supplied on the card, and the system is reset to read in the next card. This recognition program, if successful, will build up in time to a reasonable recognition rate. Incidentally, these experiments were run for zero initial memory. Thus, the recognition rate was initially zero.

In the remaining three recognition attempts, whether the number was recognized or not, the codes were always entered in the proper lists so as to prime the system to recognize each character in succeeding trials on the first recognition attempt. On the second attempt, the two codes on each card were individually compared to top and bottom codes in each of the ten list pairs. A match occurs if any of the top codes in one of the ten lists of codes matches with the top input code and if any of the bottom codes in that same list matches with the bottom input code.

If the character is not recognized on the second attempt, the first and second attempts are repeated, but with reduced codes. The top and bottom codes are reduced by removing the codes 3, 6, 9, and 12 from the original output of the line-feature generation routine if any of the codes 7, 11, 14, or 15 are present. The complicated numbers like "two" or "eight" are thus recognized on the basis of only the codes 7, 11, 14, and 15.

If the character is not recognized after the second attempt with reduced codes, the nonreduced codes are entered in the list and the next character is read. After each run, the contents of the list were punched on cards to be re-entered before the next run.

For each successive attempt of the four attempts at recognition, the ratio of recognitions to list entries improved. However, the substitution error rate, defined in terms of characters read incorrectly, increased with each attempt. These results will be discussed in the next section.

D. Results of the Simulation of Method 3

Approximately the same number of characters were written for each of the 10 numbers; 2180 characters were read in this study. Half of these characters were written on documents with boxes used as a guide, as shown in Fig. 3. Half of them were written without any guide, as shown in Fig. 4. Those written without guide boxes were run through the program before the other half.

Within each half, the characters were ordered by the number written. These ten groups were then ordered so that the first "one" written by subject A was followed by the first "one" written by subject B; this scheme was continued for all subjects before the second "one" of subject A was entered. This was done to smooth out the rate of list buildup. At the beginning of the experiment, it was noted that, on the first two attempts for each character, the system often did not recognize certain of a particular writer's numbers immediately, but required one or two samples of his writing.

The first two attempts were made from zero initial memory. The attempts with reduced codes were made after 15 "make-believe" characters (representing certain codes which occurred very frequently for each of the 10 Arabic numerals) had been entered.

Note that if an error gets into the list before it can be nullified by a correct character, a large group of char-

acters may be in error. Also, on the two attempts where individual top and bottom codes are matched, some trouble may be encountered by particular top or bottom codes if they are listed before certain pairs of codes are listed. These confusing cases were removed by making two runs and putting the troublesome characters behind the other unboxed or boxed characters on the second run. Thereupon, they were rejected as errors or entered after certain pairs of codes had been entered for other characters.

The per cent of characters not recognized and entered in the lists after each of the four attempts, plotted as a function of the number of characters read up to that point, is shown in Fig. 16. The points were obtained by averaging over 200 characters per point. Note that there is no abrupt break at 1000, which is the transition point between unboxed writing and boxed writing.

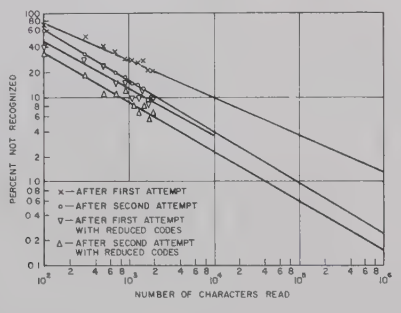


Fig. 16—Entry rates for all four attempts.

The substitution error rates with the 2180 characters for the four attempts are shown below. These do not include the three errors which were due to the inability of the logic to separate adjacent characters. The error rate shown for each attempt includes all of the errors which occurred in the previous attempts (since the logical rules for each attempt were relaxed from the previous attempts).

Error rate after:

Attempt One	0.78 per cent
Attempt Two	1.10 per cent
Attempt Three	1.46 per cent
Attempt Four	1.73 per cent.

Most of the errors were due to unresolved loops in "eights" and "nines." Other major sources were the following: "sixes" were written that looked like "eights;" "nines" were written with open tops and were confused with "fours;" "sevens" were written with closed tops and were confused with "nines."

It was feared that the error rate might increase with the number of characters entered in the list, as there is more chance for error if there are more codes listed. No significant increase in error rate was detected during the testing of 2180 characters.

Because of the limited number of samples, this was made a sequential experiment. If 1000 characters had been used to build up the list and it had been used with another 1000 characters, the result would have been a single-entry rate. However, it was felt that a sequential experiment such as this might predict the error rate for any number of characters or for any size of list. The next section of this paper discusses a model whereby the entry rate as a function of the number of characters read can be deduced. This information allows one to predict future entry rates and to predict the size of the list.

E. A Model for the List Buildup Experiment

A model of the list buildup experiment will be advanced in this section and its implications will be compared to the experimental results. Essentially the model poses the question: given a set of codes, let us pick codes at random with replacement; after the Nth such pick, what is the probability of picking a code which has not been picked before? This probability can be related to the entry rates of Fig. 16, as the codes entered are codes that were not picked before.

Assume that the codes are not equally probable but that their frequencies are given by a function w_i normalized so that

$$\sum_i w_i = 1.$$

For the Nth trial, the probability that the ith code has not been picked in $(N-1)$ trials is

$$[1 - w_i]^{(N-1)}.$$

The probability of picking the ith code is w_i, so that

$$w_i[1 - w_i]^{(N-1)}$$

is the probability of picking the ith code for the first time on the Nth try. Then the probability of picking any code for the first time on the Nth try is

$$\sum_i w_i[1 - w_i]^{(N-1)}.$$

Consider now a case where the codes have the frequency function

$$w_i = (1 - c)c^i, \qquad c < 1 \quad \text{and} \quad 0 \leq i < \infty,$$

so that c is the base of an exponential weighting. In this case, the probability P_e of picking an unpicked code is given by

$$P_e = \sum_{i=0}^{\infty} (1 - c)c^i[1 - (1 - c)c^i]^{(N-1)}.$$

The correspondence of this model to the experimental situation was checked for the unreduced recognition attempts. The frequency of occurrence of each code was plotted against its rank. This was done for individual top codes and bottom codes, and for pairs of codes for some of the numbers. It was found that an exponential frequency curve with a c value of 0.8 to 0.9 fitted the individual top and bottom code lists. The frequency curve of pair codes was much flatter than exponential.

It is simplest to apply the exponential model to the case of deriving the list-entry probability for the second

attempt. This corresponds to picking one code out of each of two lists and requiring that one of the codes has not been picked before in order to enter the code in the list. Thus if $(1-P_e)$ is the probability of a single match, the probability of a double match assuming independence is $(1-P_e)^2$. The probability of picking a pair of codes, at least one of which has not been picked before, is then given by

$$P_2 = 1 - (1 - P_e)^2.$$

In Fig. 17, curves derived for $c=0.80$, $c=0.85$, and $c=0.90$ are shown superimposed on the experimental points for the second attempt. The correspondence between the results of Fig. 16 and the theoretical curves of Fig. 17 appears to be good. Since the experimental curve is averaged over the 10 numbers, the entry rate should be dominated by the higher values of c as the number of characters read increases.

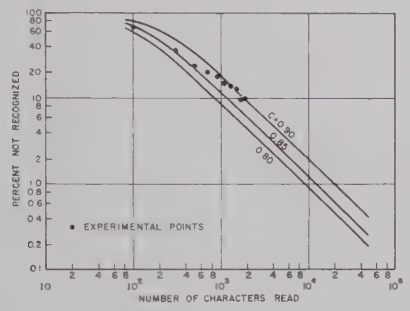


Fig. 17—Entry rate curves for the second attempt.

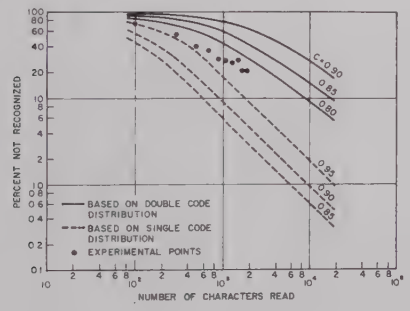


Fig. 18—Entry rate curves for the first attempt.

Let us consider a model for the first attempt. The frequency of double codes as a function of their rank starts as an exponential. Most codes occur only once, and the frequency of the most frequently occurring code is not as high as it is for the single-code frequency distributions. Fitting these data with an exponential and applying the resulting values of c to the entry rate as a function of the number of characters read yields the curves shown in Fig. 18. The theoretical curve is derived by application of the model directly to the case of a single list for each number. Curves are shown in Fig. 18 for $c=0.85$, 0.90, and 0.95. The experimental points are also shown in this figure.

If one now considers the double codes used in the first attempt to consist of pairs of independent codes distributed exponentially with values of c equivalent to those of attempt number two, a new recognition-rate curve may be derived as follows: consider a double code to consist of two independent single codes. If the single codes have frequency w_i, then the double codes must be distributed as the product of the single-code frequencies. Thus, the probability of entering a double code is given by

$$P_d = \sum_{ij} w_i w_j \left[1 - w_i w_j\right]^{(N-1)}.$$

For the exponential case with the same value of c for both codes,

$$P_d = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} (1-c)^2 c^i c^j \left[1 - (1-c)^2 c^i c^j\right]^{(N-1)}.$$

This may be reduced to a sum over a single index r by counting occurrences of power of c. This results in the probability of entry

$$P_d = (1-c)^2 \sum_{r=0}^{\infty} (1+r) c^r \left[1 - (1-c)^2 c^r\right]^{(N-1)}.$$

This expression has been evaluated for the cases $c=0.80$, $c=0.85$, and $c=0.90$, and these curves have also been plotted in Fig. 18. It is concluded that for any combination of independent or dependent individual codes distributed exponentially, the entry rate goes down with the number of characters asymptotically as some negative power of N.

The size of the list as a function of the number of characters read for each of the digits can be found by integrating the entry rate curves. This figure can be used to predict the complexity of the logic system required for recognition at a given rate or the size of memory required for a lookup-type system. The experimental results, if (the somewhat liberal) extrapolation of the curves is valid, indicate that the total list of code pairs builds up approximately as $12N^{1/2}$, where N is the number of characters recognized or entered. For the set of 2180 characters tried here, there were 778 code pairs in the ten lists. By extrapolating the entry rate curves to have the entry rate reach the error rate of 1 per cent, we find that a list size of about 10,000 entries would be required using only a machine based on the first attempt. A machine based on the second attempt would require 4000 code pairs for a recognition rate of 99 per cent. However, if only single codes which differ from previous codes are entered, rather than double codes, this list could be decreased in size considerably.

VI. Conclusion

In conclusion, it should be stated that the particular application may dictate the type of handwritten number reader to be designed. In some applications, such as that of reading telephone toll tickets,[4] any additional writing time required by telephone operators using constraint systems to prepare tickets is very expensive. Here, more operators must be hired if each operator needs more time to place and write up a call. Conversely,

the volume of tickets is so great that a very expensive recognition machine can be afforded (if accurate). Consider now an inventory application. Cards are supplied to clerks who write on these cards quantities of stock corresponding to part numbers. Here, the time required of the writer is a small part of the over-all job time. If the clerk could follow the constraints accurately, the 2-dot or 4-line constraint writing methods could be used to provide a much simpler and less expensive reading machine. What is really important to the reading machine designer is what type of control of the writer can be established and how much it will cost to read this writing using the control.

Three methods of writing control and three systems to read this writing have been presented in this paper. All of the systems described in this paper characterize rows of handwritten characters by a relatively small number of parameters. In the first two systems, a network which decodes all of the most probable states of these systems and which uses these parameters as an input can be formulated directly by the machine designer. The last system described, which was developed to read unconstrained handwritten numbers, used a computer program to construct lists of codes. A deterministic or probabilistic logic may later be developed from these lists.

Appendix

Some work has been done in characterizing the methods of generating documents used in this research. Tests were given to 20 Bell Laboratories secretaries to judge their ability to write according to the 2-dot method and to compare their writing speeds and error rates against their ordinary writing speeds and error rates. In this Appendix, a model will be indicated which helps to explain the 2-dot and 4-line controls in terms of the error rate to be expected from people as a function of their writing speed. This model will be related to the experimental results for 2-dot writing. The purpose of this section is to postulate that there may be a dexterity limit to the speed at which people can write correctly to a given constraint.

The model used here is the same as that employed by Fitts[7] and by Pierce and Karlin.[8] With this model, the human is treated as an information channel. The output from this channel involves the task of accurately tracking some visible indicia with a stylus. The inputs to this channel are visual stimuli and, to some degree, tactile stimuli.

More specifically, a subject is told to place a stylus accurately and repeatedly at a target point some fixed distance from a starting point. The subject's ability to track accurately is tested as a function of the rate of placement and as a function of the distance between his starting point and the target point. The distance between the target point and his end point is reasonably normally distributed with mean zero. For small tracking distances, this error can be related to the rate at which the task is performed by postulating analogies between length of track and signal, the error in tracking and some function of noise, and the tracking rate and bandwidth. These three factors have been related by means of a constant which has the dimensions of channel capacity.

Pierce and Karlin have derived the limiting rate for transmitting "tracking" information M:

$$M = \frac{N}{2}\log_2\left(1 + \frac{\overline{x^2}}{\overline{d^2}}\right),$$

where M is in units of bits per second; N is the number of tracking operations per second; $\overline{x^2}$ is the mean-square amplitude of track lengths; and $\overline{d^2}$ is the mean-square tracking error.

In Fitts' work and in the present study, the error rate is given in terms of the number of times that a target with a width of W is not hit. If we consider the deviation from the center of the target to have a Gaussian distribution, its mean-square error $\overline{d^2}$ can be given in terms of the error rate E and the width of the target plate W by the expression

$$E = 1 - \frac{1}{\sqrt{2\pi}}\int_{-W/2\sqrt{\overline{d^2}}}^{+W/2\sqrt{\overline{d^2}}} e^{-x^2/2}dx.$$

Let us consider Fitts' experiments. In one of these experiments, the subject was required to tap a stylus alternately on two plates, a fixed distance A apart. These plates were long and narrow with the narrow dimension W along the line of the track. The subject was told to strike the two target plates alternately and to score as many hits as he could. If the subject missed either of the plates, these errors were recorded. The widths of the plates and the distance between them was varied. Values of M have been calculated using Fitts' data. The results of Fitts' study are summarized in Table II.

TABLE II
Stylus Tracking Data

←——— A ———→ →|W|←				→|W|←
W inches	A inches	Time sec	Error per cent	Rate M bits/sec
0.25	2	0.392	3.35	13.0
0.25	4	0.484	3.41	12.6
0.25	8	0.580	2.78	12.3
0.25	16	0.731	3.65	11.1
0.50	2	0.281	1.99	15.0
0.50	4	0.372	2.72	13.8
0.50	8	0.469	2.05	13.3
0.50	16	0.595	2.73	12.0

[7] P. M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement," *J. Exp. Psychol.*, vol. 47, pp. 381–391; June, 1954.

[8] J. R. Pierce and J. E. Karlin, "Reading rates and the information rate of a human channel," *Bell Sys. Tech. J.*, vol. 36, pp. 497–516; March, 1957.

This same model for tracking will now be applied to the two constrained writing systems studied in this paper. We will consider each of the digits written to consist of a given number of such tracks. The objective for the writer is, starting at a given point, to move a stylus to another point along a number without making an error. An error in the 2-dot system is considered to result when the stylus ends on the wrong side of a dot, outside a guide line, or in the space assigned to the adjacent character. In the 4-line constraint system, an error is considered to occur if the end of a track lies outside the two sets of upper and lower constraint lines. A model will be considered where the initial track to the beginning of a character is ignored.

Consider the tracking task of writing correctly to the 2-dot constraint the example "two" shown in Fig. 19. The writer, after starting the stylus in the upper left-hand box, must place it accurately in the upper right-hand box. If the stylus lands outside of that box, an error is scored. In this simplified model, the "two" is considered to consist of five such tracking tasks as shown by the 5 segments of the number "two" in Fig. 19. Each of these tracking tasks corresponds (ignoring size) to the task required in Fitts' experiments. Thus, an error rate can be obtained in terms of the distance tracked, the dimensions of the target areas, the total number of tracks made per unit time, and a constant having the dimensions of channel capacity. The error rate as a function of writing speed, so calculated, should be a lower bound.

In this simple model, all strokes are considered to be of the same length; the target width is considered to be one half the tracking distance for all strokes. The total number of strokes used in this model for the 10 numbers was 40 (or 4 per character). In Fig. 20 are shown experimental curves of the error rate (as defined above) as a function of writing speed in writing with the 2-dot constraint and in writing with no constraint. These curves were obtained from a series of tests given to 20 secretaries after they had reached a learning plateau for this task. The model curve was adjusted to be tangent to the experimental curve by varying M. This curve is also plotted in Fig. 20. The value of M found from the model was 12.3 bits per second, which agrees well with Fitts' results.

This model has been introduced in support of the claim that there is indeed a limit to the tracking ability of people. This may be called a dexterity limit in writing under a constraint system. It is interesting to speculate why the experimental curve deviates so greatly from the theoretical curve and in fact levels off at low writing speeds. One may suppose that at low writing speeds, people did not take the necessary time to form their characters correctly. In effect, they were loafing and could have done better.

The same type of model can be applied to the 4-line writing constraint system. In this system, one need only consider vertical strokes in each of the 10 numbers. The number of such strokes is found to be 21. In a first approximation, it is reasonable to assume that the tracking distance and target-size relationships are the same as that of the 2-dot system. That is, the tracking distance is twice the error distance. It can then be concluded that if the subjects write as efficiently as possible, they should write under the 4-line constraint about twice as many characters per unit time as under the 2-dot constraint. However, it is found that such a rate would be nearly on the curve for ordinary writing without any constraint. It is felt that here the stylus speed as well as the speed of initial placement must be accounted for. Only very limited tests were made of the 4-line type of constraint, but an improvement over the 2-dot system was indicated.

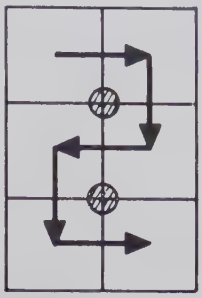


Fig. 19—Tracking a "two."

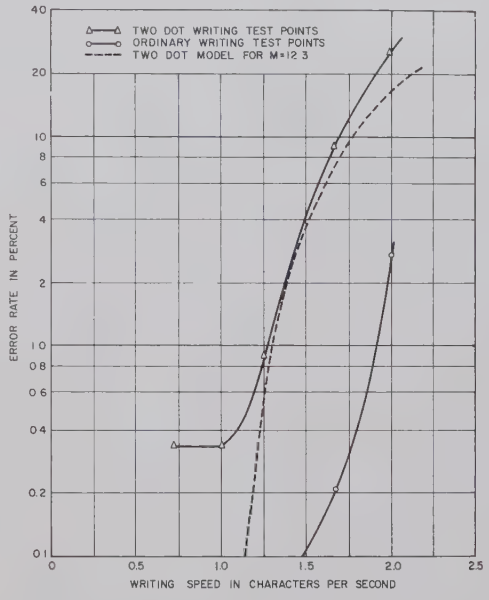


Fig. 20—Error rate vs writing speed.

Acknowledgment

The author would like to acknowledge the assistance given by certain individuals, all of the Bell Telephone Laboratories: A. C. Mehring and T. L. Dimond for their support of this project; W. H. Highleyman for his help in getting into operation the scanner which made this study possible; G. Gordan for writing the scanner tape generation program; and E. Wolman for his advice on the list buildup model.

An Analog Method for Character Recognition*

W. H. HIGHLEYMAN†, MEMBER, IRE

Summary—A method for character recognition which is capable of an analog implementation has been studied by simulation on a digital computer. In essence, this method involves maximizing the cross-correlation value between the unknown character and a set of average characters, there being one average character for each allowed character class. An average character is represented by a two-dimensional function. The value of this function at a point is the probability of occurrence of a mark at that point for the character class represented by the average character. Negative weights are given to areas of low probability in each average character to improve discriminability.

The simulation results indicate that this method is applicable to the recognition of machine printing, and perhaps to the recognition of constrained hand printing. The method can be implemented in an economical manner using electro-optical techniques.

Introduction

BY AND LARGE, the pattern-recognition machines of today, whether in use or merely proposed, are digital in nature. That is, the pattern to be recognized is generally quantized in both position and density before any of the procedures for recognition are applied. The recognition procedure is generally then implemented by using binary logical circuitry. There are several advantages that might be gained if the pattern and all pertinent or derived information is kept in an analog form[1,2] for as long as possible. Prominent among these advantages are low cost and lack of quantizing error. In addition, a speed advantage may sometimes be realized.

A particular method for character recognition which is capable of an analog implementation has been studied by simulation on the IBM 704 digital computer. The results of the simulation indicate that this method is applicable to the recognition of machine printing, and perhaps to the recognition of constrained hand printing. The method can be implemented with simple optics for the most part, yielding an economic machine.

This method of character recognition is described in this paper. Simultation parameters and results are presented, and a means for optically implementing the method is discussed.

Description of Method

The character recognition method to be described depends upon a set of average characters, there being one average character for each of the allowable character classes. (A character class is the collection of all the symbols that are identified as a particular character.) For instance, if the allowable input characters are the alphabetics A through Z, then there will be an average character for an A, one for a B, etc. An unknown character is identified by comparing it to this set of average characters and determining that average character to which it most closely corresponds (the measure of correspondence will be defined below).

The unknown character is represented by the distribution of marks in two dimensions. An average character is likewise represented by a two-dimensional function. The value of this function at a point is the frequency of occurrence of a mark at that point computed over the character class of the average character. Fig. 1 shows an example of functions representing an average character and an unknown character.

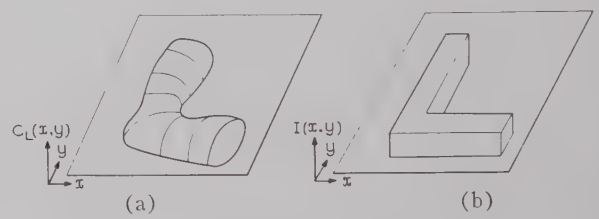


Fig. 1—Two-dimensional functions representing: (a) An average L. (b) An input (unknown) character (an L).

The measure of correspondence between an unknown character and a particular average character is the cross-correlation value between the two. An unknown character is identified with that character class represented by the average character with which the greatest cross-correlation value is obtained. Since this is a position-sensitive identification criterion, the unknown character must be shifted in two dimensions with respect to each average character. A cross-correlation function (a function of this two-dimensional shift) is computed between the unknown character and each average character. The maximum of each such function is chosen to represent the correspondence between the unknown character and that average character. The absolute maximum of these local maxima then forms the recognition criterion.

The above statements are formulized below:

The cross-correlation function between the unknown character and the jth average character as used here can be defined as[3]

* Received by the PGEC, December 8, 1960.

† Bell Telephone Labs., Inc., Murray Hill, N. J.

[1] K. R. Eldridge, F. J. Kamphoefner, and P. H. Werdt, "Automatic input for business data-processing systems," *Proc. EJCC*, New York, N. Y., December 10–12, 1956, pp. 69–73.

[2] W. K. Taylor, "Pattern recognition by means of automatic analog apparatus," *Proc. IEE*, vol. 106, pt. B, pp. 198–209; March, 1959.

[3] C. K. Chow, "An optimum character recognition system using decision functions," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-6, pp. 247–254; December, 1957.

$$\Phi_j(\sigma, \rho) = \frac{\int_x \int_y I(x+\sigma, y+\rho) C_j'(x, y) dxdy}{\left[\int_x \int_y I^2(x, y) dxdy \int_x \int_y C_j'^2(x, y) dxdy\right]^{1/2}}, \quad (1)$$

where

$C_j'(x, y)$ = the two-dimensional function representing the average character,

$I(x+\sigma, y+\rho)$ = the two-dimensional function representing the unknown (Input) character, shifted with respect to $C_j'(x, y)$ by distances σ, ρ in the x, y directions,

$\Phi_j(\sigma, \rho)$ = the cross-correlation function between $C_j'(x, y)$ and $I(x+\sigma, y+\rho)$, as a function of the two-dimensional shift, σ, ρ,

$\int_x \int_y (\ ) dxdy$ = integral over the two-dimensional character field.

The integral $\int_x \int_y C_j'^2(x, y)\ dxdy$ is the norm of $C_j'(x, y)$. If $C_j'(x, y)$ is normalized by the square root of its norm, then the resulting function $C_j(x, y)$ is

$$C_j(x, y) = \frac{C_j'(x, y)}{\left[\int_x \int_y C_j'^2(x, y) dxdy\right]^{1/2}}, \quad (2)$$

and

$$\int_x \int_y C_j^2(x, y) dxdy = 1. \quad (3)$$

Then,

$$\Phi_j(\sigma, \rho) = \frac{\int_x \int_y I(x+\sigma, y+\rho) C_j(x, y) dxdy}{\left[\int_x \int_y I^2(x, y) dxdy\right]^{1/2}}. \quad (4)$$

For the remainder of this paper, the average character function, $C_j(x, y)$, will always be assumed to be normalized so that (3) holds.

Note that the norm of $I(x, y)$, $\int_x \int_y I^2(x, y)\ dxdy$, is common to all $\Phi_j(\sigma, \rho)$ for a particular input pattern; hence, neglecting it causes no reordering of the $\Phi_j(\sigma, \rho)$. Hence, maximizing the modified cross-correlation function $\Phi_j'(\sigma, \rho)$, given by

$$\Phi_j'(\sigma, \rho) = \int_x \int_y I(x+\sigma, y+\rho) C_j(x, y) dxdy, \quad (5)$$

is an equally valid recognition criterion. The true correlation value given by (4) will be used in the description of the simulation study of this method so that comparisons between recognition attempts can be easily made. However, the simplified form (5) will be used to advantage in the discussion of the optical implementation.

A modification to the $C_j(x, y)$ was studied. This is the addition of penalty areas to the average characters. A penalty area is an area of low probability to which a negative penalty weight is assigned. In each average character, the penalty weight for all penalty areas is a constant. It is arbitrarily chosen so that the integral of the penalty weight over all penalty areas in a given average character is unity.

Although the final machine using this method does not necessarily require quantization, the simulation of the method on a digital computer does. For purposes of simulation, an unknown character is represented by a 12×12 matrix of ones and zeroes. (This is both a spatial and a mark-intensity quantization.) A one corresponds to a mark in an element, a zero to no mark. The decision concerning the presence of a mark is based upon an appropriate threshold level. Each average character is also represented by a 12×12 matrix, with the value assigned to each element being proportional to the probability of occurrence of a mark in that element. The integrals in (3), (4), and (5) must then be replaced by the appropriate sums:

$$\Phi_j(\sigma, \rho) = \frac{\sum_m \sum_n I_{(m+\sigma),(n+\rho)} C_{jmn}}{\left[\sum_m \sum_n I_{mn}^2\right]^{1/2}}, \quad (6)$$

$$\Phi_{j'}(\sigma, \rho) = \sum_m \sum_n I_{(m+\sigma),(n+\rho)} C_{jmn}, \quad (7)$$

where

$$\sum_m \sum_n C_{jmn}^2 = 1. \quad (8)$$

The notation is the same as that used previously, except that the discrete subscripts m, n replace the continuous variables x, y. σ and ρ are also discrete in this case. Note that $\Phi_j'(\sigma, \rho)$ in the quantized case above is simply the sum of the weights of the marked elements of C_j, since each I_{mn} can only be zero or one. Likewise, the norm of I, $\sum_m \sum_n I_{mn}^2$, is simply the sum of the marked elements in the matrix representing the input character.

A simple example will illustrate the mechanics of this method. Assume that the unknown and average characters are represented by 3×3 matrices. Fig. 2(a) and (b) shows hypothetical unnormalized and normalized average characters (hereafter called probability matrices) for a C and an 0. Penalty weights have been added to areas of near-zero probability. An input character (an 0) is shown in Fig. 2(c). Shifting is not performed in this simple example since the optimum positions are obvious. The pertinent modified correlation values are shown, the maximum of which clearly identifies the input character properly.

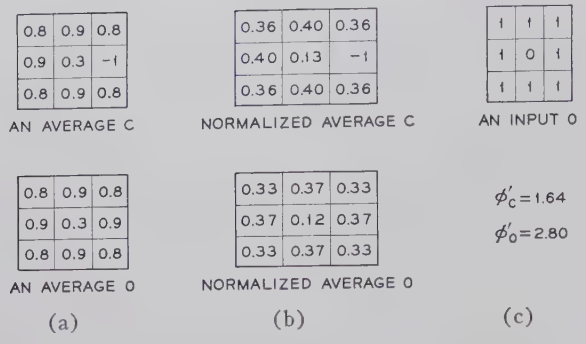


Fig. 2—A recognition example.

Simulation

This character recognition method was studied by simulation on the IBM 704 computer. As previously discussed, the characters were represented by 12×12 matrices of one's and zero's. This degree of quantization introduced a quantizing error affecting the results, but was necessary to maintain reasonable computer time.

The method was applied to two different sets of data: a set of hand-printed alphanumeric characters and a set of machine-printed numbers. The hand printing consisted of 1800 characters (50 alphabets of the 36 alphanumeric characters) printed by 50 different people. This printing was somewhat constrained by requiring the writer to print on one-quarter inch quadruled paper, asking him to print neatly and at a size approximating the ruled boxes on the paper.

The source of the machine printing was an IBM 407 line printer. 1000 numbers were studied representing 100 samples of each of the ten numerals. These were taken from 80 different type wheels.

The samples were scanned and converted to matrix form by the use of the generalized scanner,[4] an optical scanner that can be programmed to generate any type of scan within its resolution capabilities. The scanner output is a magnetic tape compatible with the IBM 704 computer.

The primary value of the study of hand printing was the sensitivity of the recognition results to various parameters (such as various methods of centering, penalty area variations, etc.). Since the hand-printed data was not as "handicapped" by high percentage recognition as the machine-printed data (recognition rates for the former were in the order of 60–80 per cent), this data furnished a sensitive test to evaluate variations in the recognition methods.

The particular parameter ranges which were determined to be best were then applied to the recognition of machine-printed characters.

Parameters of Investigation

One parameter of the simulation study was the method of centering. Two methods were investigated: centering by center of gravity alignment, and centering by maximizing the cross-correlation value as a function of position (centering by shifting). In the former, the center of gravity of the input pattern is aligned with the center of gravity of the probability matrix (average character). In the latter method, the input pattern is shifted in two dimensions with respect to each probability matrix, and the cross-correlation value is computed for each position. Hence, the correlation obtained with a particular probability matrix is a function of input pattern position. The maximum value of this correlation function is used as the measure of fit between the input pattern and that probability matrix. Although centering by shifting seems to be intuitively better, center-of-gravity centering has some advantages. For one, it requires much less computer time for simulation. In addition, there was the possibility that centering in this manner would eliminate the tendency of a character to find a false maximum correlation, when compared to a probability matrix other than its own, by finding some opportune misalignment.

A second parameter involved the question of penalty weights. Penalty weights are negative numbers assigned to elements of low probability. Hence if an unknown character falls in a region of low probability with respect to a particular probability matrix, then the corresponding cross-correlation value is reduced, or "penalized." Penalty weights then have the possibility of increasing the discrimination between characters. In this study, a penalty threshold level was chosen so that any elements with a probability less than the penalty threshold would be assigned the penalty weight. All penalty elements were assigned identical weights, and these weights in each matrix were arbitrarily normalized so that the sum of the squares of the weights was unity (the same normalized value of the matrix) as previously discussed. The value of the penalty threshold level was varied to determine the effect on error rate. The effect of nonuniform penalty weights and the effect of other normalizing criteria were not studied.

The third parameter studied was that of rejection criteria. Here we are interested in setting certain criteria for the final cross-correlation values in order that the recognition be acceptable. If the recognition is not acceptable, the character is rejected as being unreadable. Through the use of rejection criteria, the undetected error rate (substitutional errors) can be made as small as desired by making the rejection rate as large as necessary. The particular rejection criteria considered required that the maximum correlation value exceed a particular threshold level and, further, that it be greater than the next highest correlation value by a prescribed discrimination level.

In summary, then, the parameters of this study included:

1) Centering methods.
2) Penalty areas.
3) Rejection criteria.

[4] W. H. Highleyman and L. A. Kamentsky, "A generalized scanner for pattern- and character-recognition studies," *Proc. WJCC*, San Francisco, Calif., March 3–5, 1959; pp. 291–294.

Construction of Probability Matrices

An unnormalized probability matrix for a particular character is constructed by determining the probability of occurrence of a mark in each of the elements of the matrix when the input pattern is that character. In this study, 50 samples of the pertinent character were used to construct each probability matrix. Different probability matrices were, of course, used for the machine printing and for the hand printing.

Since there is no mechanism for properly centering the characters in the matrix scanning process, care must be taken to assure that they are properly centered before constructing the probability matrix. First, though, one must decide just what constitutes proper centering. Since all of the recognition methods concerned maximize a function which is monotonically increasing with the cross-correlation function, it seems reasonable to define proper centering of an input pattern with respect to the probability matrix as that position which maximizes the cross-correlation function between the two.

However, in initially constructing the probability matrices, there exist no such matrices which can be used to center the patterns. Therefore, the process of construction must be an iterative one. Considering the case of a particular character, the first step is to construct a probability matrix for that character from the unshifted sample members. Then the cross-correlation function (as a function of two-dimensional shift of a maximum of ± 5 elements in each direction, or 121 positions) is computed for each sample member compared with the first probability matrix, and its optimum position with respect to the first probability matrix is determined by the maximum of the correlation function. When all of the optimum positions of the sample members have been found, they are shifted to these positions, and a new probability matrix is constructed. This process is repeated between the sample members and the probability matrix until the elements of the probability matrix converge to their final values. The IBM 704 computer was utilized to carry out the iterations.

It seems reasonable that a test of convergence of the elements of such a matrix might be the auto-correlation value (the sum of the squares of the probabilities) of that matrix. That is, the final probability matrix is that matrix which maximizes the cross-correlation values of all of the component matrices (the sample members) with itself; therefore, one might expect that this also maximizes the auto-correlation value of the probability matrix.

The auto-correlation value was used to test the convergence of this iteration process. It was indeed a valid test, as most of the matrix elements for the hand printing converged after seven or eight iterations, and the machine printing matrix elements converged after three or four iterations. Each iteration took 30 seconds on the IBM 704 computer.

Fig. 3 shows some typical convergence curves for some of the characters. Figs. 4 and 5 show examples of the first and final unnormalized probability matrices for a hand-printed character and for a machine-printed character. The numbers in these matrices are the actual number of sample members which contained a mark in that element; division by 50 yields the probability of occurrence of a mark.

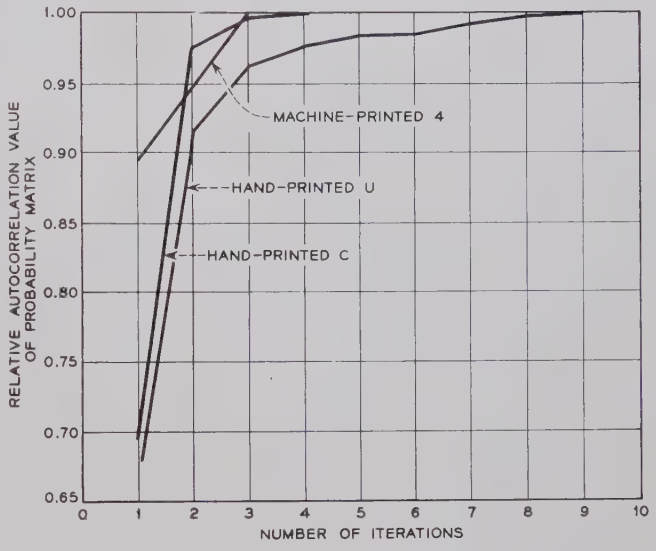


Fig. 3—Convergence curves for probability matrix iteration.

0	0	0	0	1	2	3	3	2	1	1	0
0	0	2	2	8	11	12	12	6	1	1	0
0	1	7	14	25	32	32	33	21	6	3	1
0	6	20	21	26	16	11	16	13	6	2	0
3	15	25	22	11	6	2	3	4	3	2	0
7	25	24	12	2	1	1	0	0	0	0	0
5	20	30	12	1	1	1	0	3	2	2	0
6	16	19	12	15	15	14	17	13	6	3	1
4	9	14	15	17	15	17	16	12	4	3	0
2	3	7	7	13	10	8	6	6	5	0	0
2	0	1	2	4	4	5	4	1	1	0	0
1	0	1	1	3	1	1	0	0	1	0	0

(a)

0	0	0	0	1	0	1	0	2	0	1	0
0	0	1	1	1	9	18	18	9	4	3	2
0	0	1	7	37	44	39	32	23	6	2	2
0	0	18	46	30	6	8	10	9	8	2	1
0	8	47	20	0	1	3	1	2	4	0	0
0	27	41	3	0	1	0	0	0	0	0	0
0	28	30	2	1	1	3	2	2	1	2	0
0	16	26	15	10	9	11	8	7	7	4	1
0	5	16	23	25	25	24	23	16	9	4	0
0	1	2	10	11	10	10	8	6	2	0	0
0	0	0	2	1	2	2	1	2	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

(b)

Fig. 4—(a) Initial probability matrix for hand-printed *C*. (b) Final probability matrix for hand-printed *C*.

0	0	0	0	32	32	0	0	0	0	0	0
0	0	3	25	39	24	3	0	0	0	0	0
0	1	13	38	33	8	2	8	1	0	0	0
0	5	36	41	12	2	15	24	8	0	0	0
3	19	43	31	2	0	21	43	13	0	0	0
22	44	49	31	21	19	36	48	37	21	11	0
28	47	49	45	47	46	48	50	47	46	32	0
7	19	17	18	24	19	37	49	34	20	5	0
0	1	0	2	2	1	19	46	23	1	0	0
0	0	0	0	0	0	3	15	6	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

(a)

0	0	0	0	30	38	1	0	0	0	0	0
0	0	0	25	49	22	0	0	0	0	0	0
0	0	9	43	40	1	0	10	2	0	0	0
0	1	39	50	6	0	9	30	11	0	0	0
2	20	50	30	0	0	13	49	14	0	0	0
24	49	50	26	22	21	40	50	36	22	16	0
35	50	50	50	50	50	50	50	50	50	33	0
6	13	11	16	20	16	32	50	37	13	1	0
0	0	0	0	0	0	13	48	25	0	0	0
0	0	0	0	0	0	3	13	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

(b)

Fig. 5—(a) Initial probability matrix for machine-printed 4. (b) Final probability matrix for machine-printed 4.

Recognition of Hand Printing

As mentioned previously, the primary purpose of studying hand-printed characters was to determine the effects of the various parameters. It was found early in the study that the use of all of the 1800 hand-printed characters for all of the tests was prohibited by the computing time required. For instance, a recognition trial using these characters in which centering is accomplished by shifting the input pattern a maximum of two elements in each direction required six hours of 704 time. Therefore, the determination of the effect of these parameters was deduced from just the hand-printed numbers (the same 500 characters which were used to construct the probability matrices). To process this subset of the hand-printed characters required about fifty minutes of computer time under the above conditions. The optimum values thus found were then applied to the total set of hand-printed characters and to the machine-printed numbers.

The graph of Fig. 6 presents the error rates as a function of the method of centering and the penalty criteria for the hand-printed numbers.

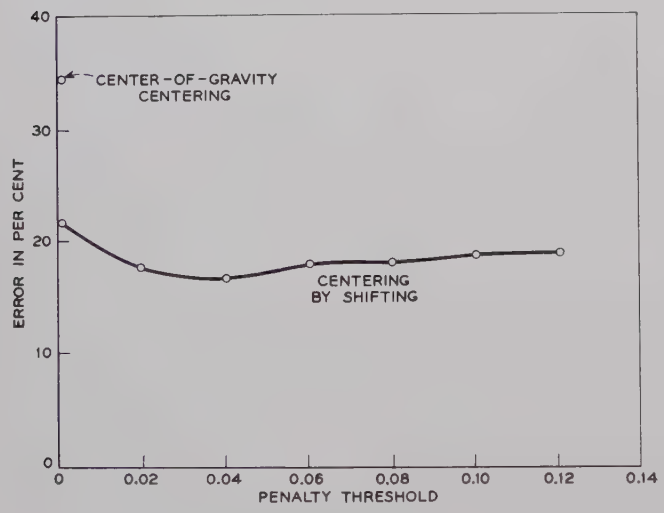


Fig. 6—Error rate for various parameters using hand-printed numbers (500 samples).

Penalty Criteria

The abcissa of Fig. 6 represents the penalty threshold used in the various tests. Each element with a probability *less than* the penalty threshold is assigned a penalty weight. Hence, the ordinate axis of the graph corresponds to error rates in which penalty areas were not used. As discussed earlier, these penalty weights are negative numbers which are constant in each matrix, and which are adjusted so that the sum of their squares in a particular matrix is unity.

In Fig. 6 the effect is shown of the penalty threshold on the per cent error. Note that the error is a minimum for a penalty threshold of 0.04.

Centering

The two methods of centering which were studied were described earlier. The particulars of centering by shifting warrant comment. In this case, to minimize simulation time, the input pattern was first roughly aligned with a probability matrix by using center-of-gravity alignment. Then the input pattern was shifted a maximum of two elements in all horizontal and vertical directions (25 positions). The maximum value of of the correlation function (a function of position) was chosen to represent the degree of match between the input pattern and that probability matrix to which it was being compared.

The two methods of centering were studied for the case of zero penalty threshold. The results are shown in Fig. 6, in which it is seen that centering by shifting is significantly better. Hence, whatever advantages center-of-gravity centering might have had (as discussed earlier) were not borne out by these results.

Rejection Criteria

The two rejection criteria studied, in review, are criteria applied to the resulting correlation values which determine an acceptable recognition. One criterion is a threshold level below which a score is rejected. The other is a discrimination level which requires that the top score and the next highest score be separated by a certain amount. The effect of various recognition criteria on hand-printed numbers was studied in detail for the case of a penalty threshold of 0.1 and centering by shifting.

Fig. 7(a) shows the dependence of the over-all rejection rate on the threshold level (T) and the discrimination level (D). Fig. 7(b) illustrates the dependence of the undetected (substitutional) error rate on the rejection parameters (the per cent undetected error rate is the per cent of the whole sample).

It is of interest to plot the loci of constant undetected error rate on these graphs so that the rejection rate which is required to achieve a desired maximum undetected error rate can be discovered. In Fig. 7(b), these loci are simply horizontal lines. Some loci for particular error rates are shown dotted. The intersections of these loci with lines of constant D can be used to plot similar loci on the graph of Fig. 7(a), where they are shown again with dotted lines.

The interesting result of this construction, as seen from Fig. 7(a), is that the minimum rejection rate for a prescribed error rate occurs for $T=0$ and D a particular value. That is, for this particular set of samples, the threshold level is meaningless as a criterion for rejection. Evidently, characters with very low maximum correlation values should still be accepted as long as the difference between the highest and next highest correlation value is sufficient.

Using, then, the optimum values of the recognition

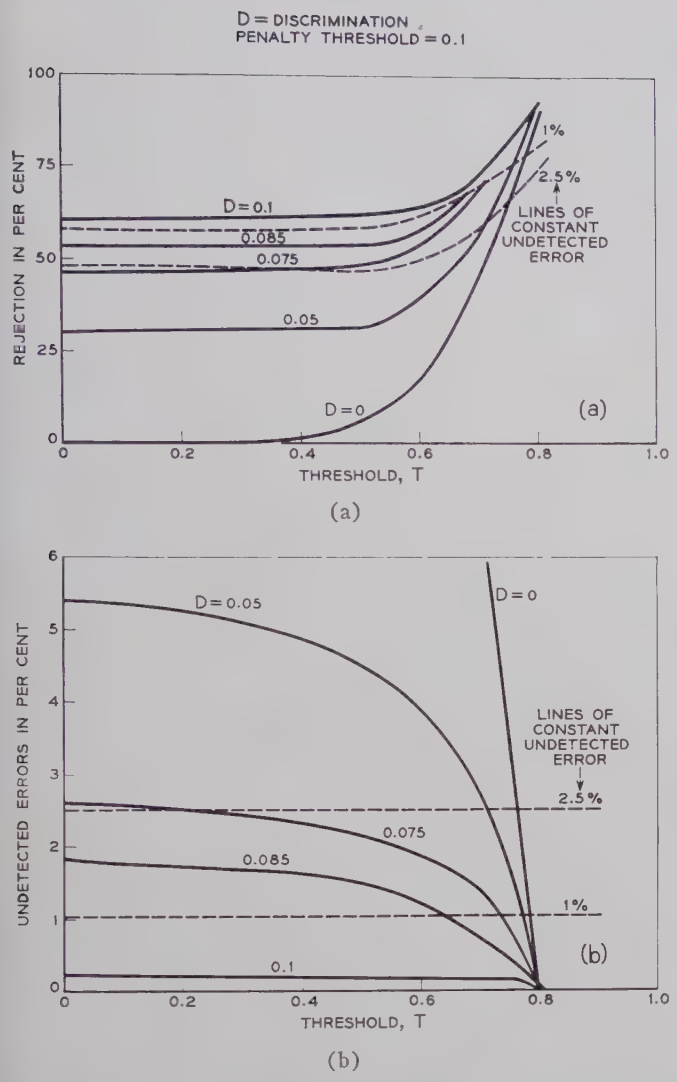


Fig. 7—Per cent rejection and per cent undetected errors as a function of the rejection parameters. D = discrimination. Penalty threshold = 0.1.

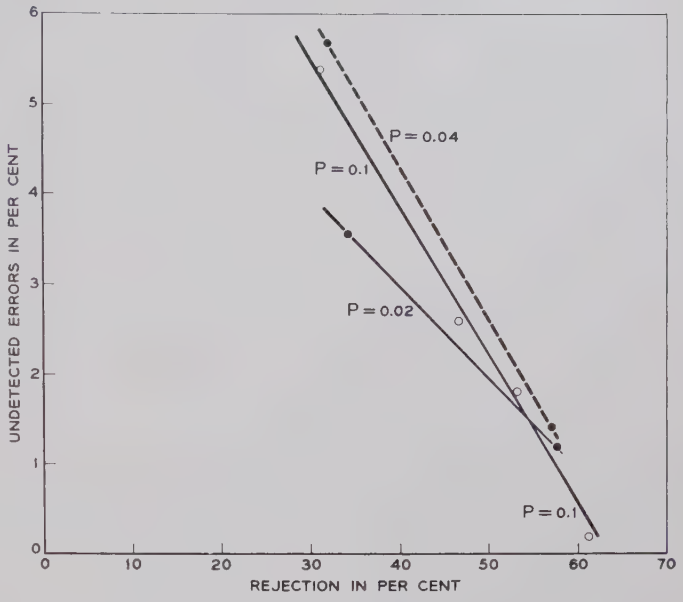


Fig. 8—Undetected error rate vs rejection rate for hand-printed numbers. P = penalty threshold.

criteria (*i.e.*, $T=0$), the dependence of undetected error rate on rejection rate can be determined. This relation, for a penalty threshold of 0.1, is shown in the graph of Fig. 8 by the points enclosed in circles. Note that the resulting curve is approximately a straight line in the region considered.

Assuming then a linear dependence between error rate and rejection rate, similar curves for other penalty thresholds were determined. Shown in Fig. 8 as a heavy line are those particular values which give a minimum rejection rate for a particular error rate, and also the curve for a penalty threshold of 0.04. It was this latter value that yielded the lowest error rate before rejection criteria were applied (see Fig. 6).

Some interesting points can be noted from Fig. 8:

1) The relation between undetected error rate and rejection rate for a given penalty threshold is approximately linear in regions of low error rate, as previously noted.
2) The penalty threshold value for which the lowest real error rate is attained is not necessarily optimum when one considers rejection criteria.
3) The particular penalty threshold which is optimum when rejection criteria are used is a function of the undetected error rate desired.

Examples of Hand-Printed Samples

Although the primary purpose of studying hand printing was to ascertain the effects of certain parameters on the recognition ability of this method, it is of interest to determine just how well the optimum parameters would do on hand printing. Consequently, this method, with centering by shifting and a penalty threshold of 0.04, was applied to the hand-printed alphanumeric alphabet of 1800 samples. No rejection criteria were applied. The total recognition rate was 77.2 per cent.

In Fig. 9 are shown some of the actual input data used. Shown in Fig. 10(a) are some matrix forms of high quality and degraded characters which were read correctly. In Fig. 10(b) are matrix forms of some characters read incorrectly. The entries beneath the matrices in Fig. 10(a) show the first and second choices and their correlation values (the first choices are all correct). Below the matrices in Fig. 10(b) are the first choices and their correlation values, as well as the actual identity of the characters with their associated correlation values.

A note of caution is necessary here. These results for the hand printing are based on the same data which were used to determine the probability matrices. It is quite doubtful that 50 alphabets comprise a large sample of hand-printed characters. Hence, one would expect a

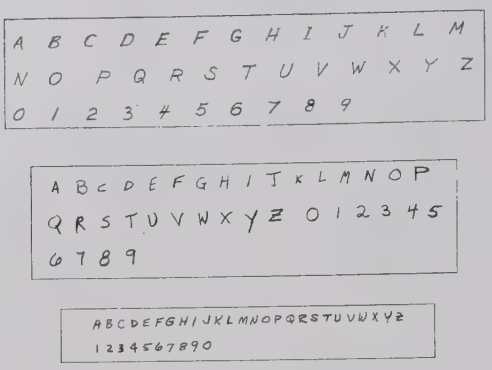


Fig. 9—Some samples of the hand-printed data.

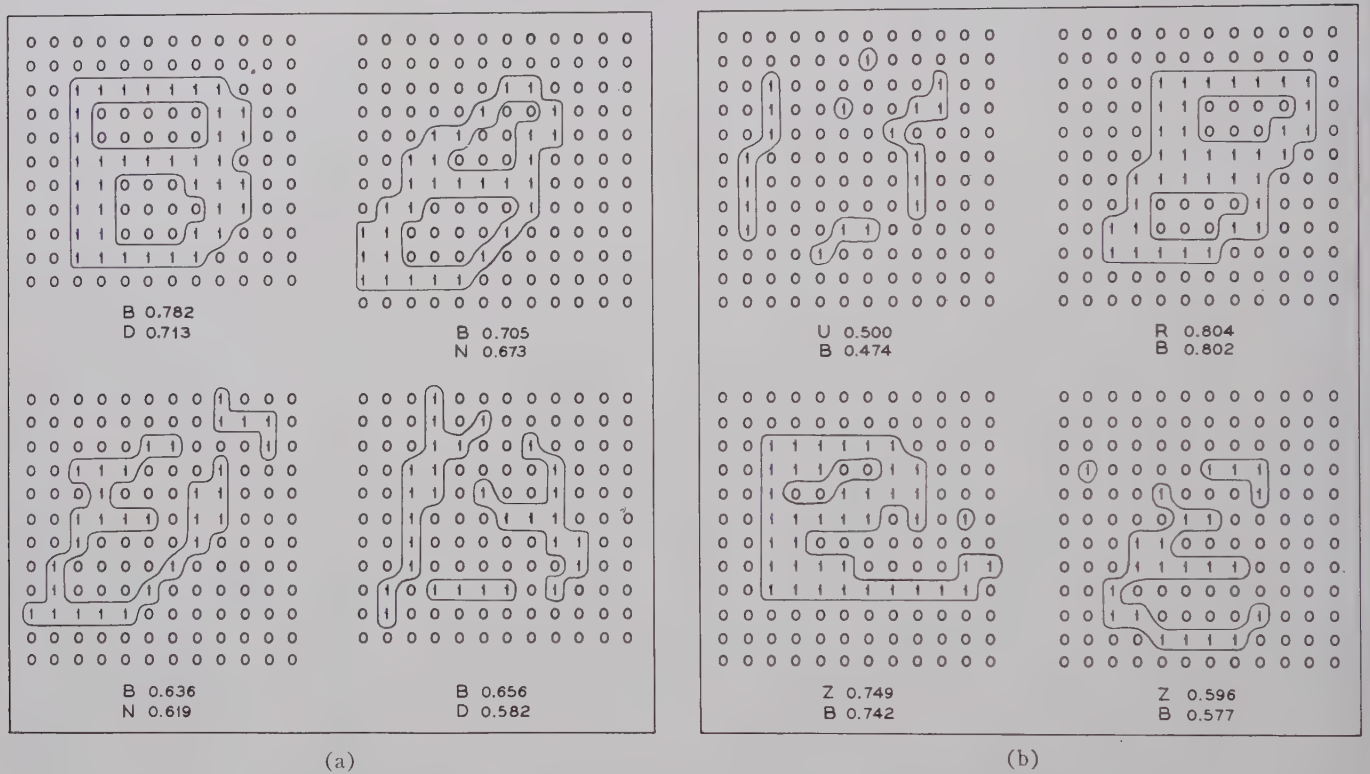


Fig. 10—(a) Hand-printed *B*'s recognized correctly. (b) Hand-printed *B*'s recognized incorrectly.

Fig. 11—Sample of IBM 407 printing.

significantly lower recognition rate for input characters other than the original data. However, it is possible that sufficient writing constraints on the originator may exist which would yield a usable recognition rate with this method. A finer quantization might also give some improvement.

Machine Printing

Parameters: As a consequence of the above investigation, recognition with centering by shifting was applied to the 1000 machine-printed numbers, a sample of which is shown in Fig. 11. The penalty threshold was varied from 0 to 0.06 to find an optimum value, since there was no optimum value clearly indicated by the previous investigation. The minimum rejection criteria required to detect all errors was applied. In this case, the number of errors was so small that a perusal of the data showed that a rejection threshold T of zero was still optimum.

Results

The results of using the above parameters are shown in Fig. 12. The discrimination level D ($T=0$) required at each point is shown. It is clear from this graph that

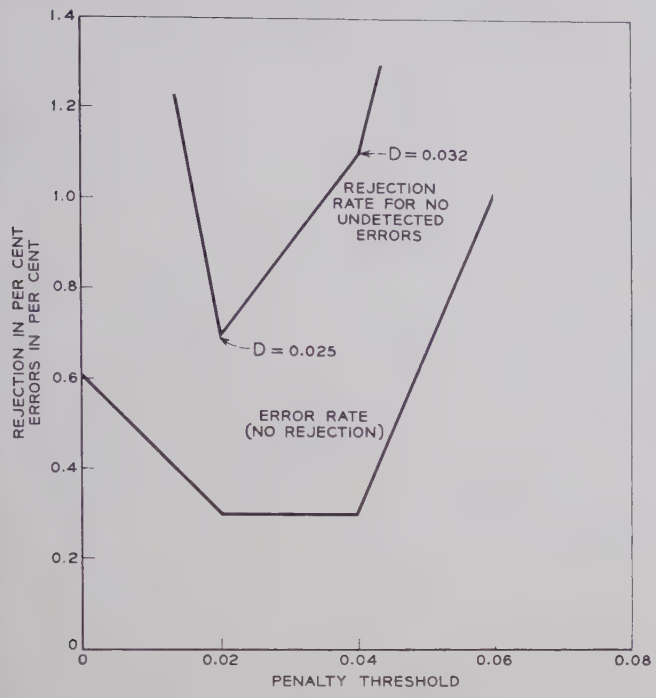


Fig. 12—Error and rejection rates for machine-printed numbers. D = discrimination level.

the optimum penalty threshold is 0.02, for which a rejection rate of 0.7 per cent guarantees no errors. Without rejection, the error rate is 0.3 per cent at this point.

Of the set of 1000 machine-printed members studied, 500 were used to construct the probability matrices. Since one might expect these characters to do better in the recognition process than the remaining 500, it is of interest to compare the results of the two sample subsets. For the subset used to construct the probability matrices, the per cent error was 0.2 per cent and the per cent rejection was 0.8 per cent. For the remaining samples, the per cent error was 0.4 per cent and the per cent rejection was 0.6 per cent. Since there is little difference between these results, the results of either subset or of the complete set should be valid.

Error Analysis

Because of the small number of errors in the optimum case for machine printing, each one can be examined in detail. The matrix forms of some normal characters are shown in Fig. 13. In Figs. 14 and 15 are shown the seven rejected numbers, the ones in Fig. 14 being the ones which were incorrectly recognized. Below these matrix forms are correlation values similar to those of Fig. 10.

The errors in Figs. 14(c) and 15) are explainable as centering or quantizing errors. The reasons for the errors shown in Fig. 14(a) and (b) are not clearly understood.

Fig. 16 (next page) shows some degraded characters which were recognized correctly, along with the first and second choice correlation values.

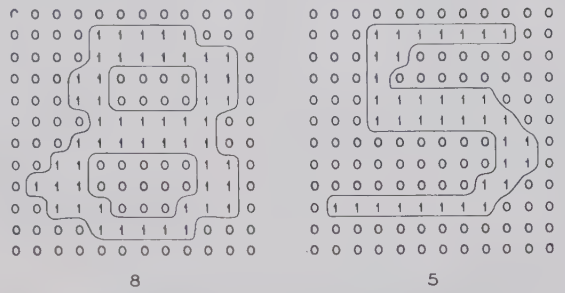
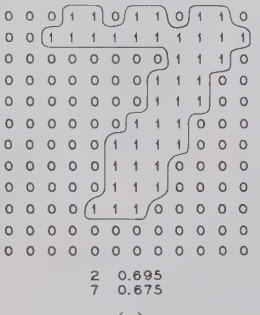


Fig. 13—Examples of normal machine-printed characters.



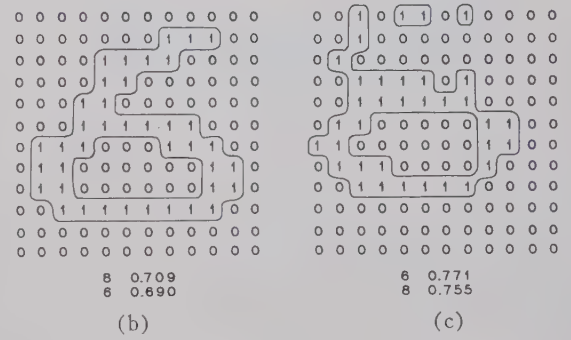


Fig. 14—Machine-printed numbers recognized incorrectly.

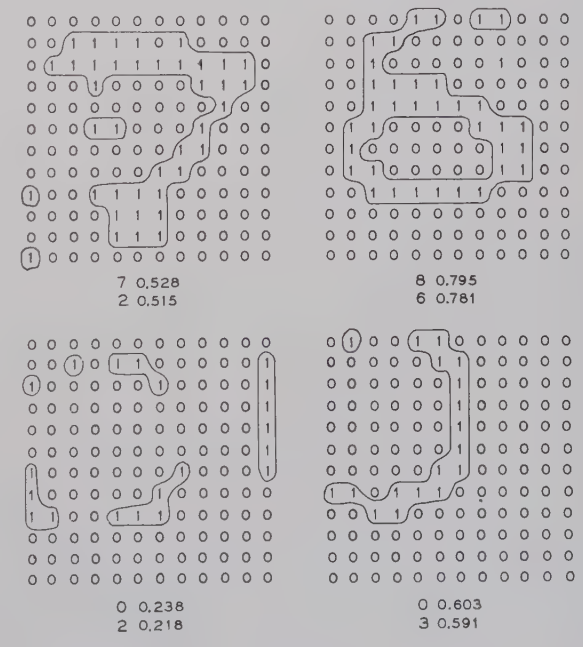


Fig. 15—Machine-printed numbers recognized correctly but rejected along with the numbers of Fig. 14.

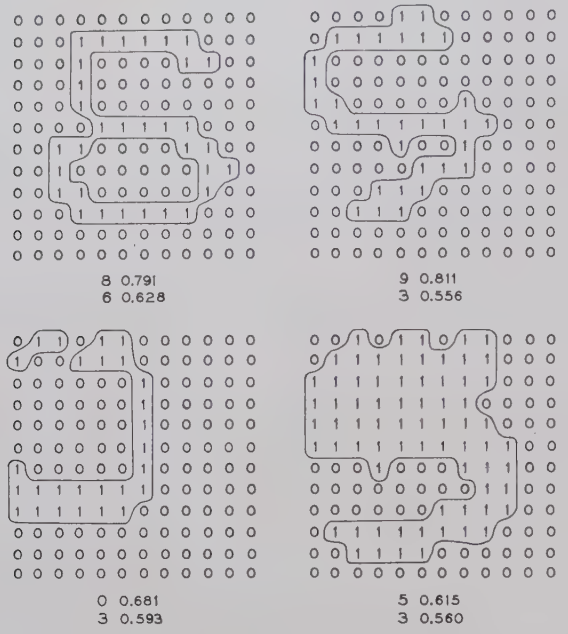


Fig. 16—Some degraded machine-printed numbers recognized correctly.

Summary of Results

The important results of this study are summarized in Table I. It seems that this method should be applicable to the reading of machine-printed characters, and that, with proper engineering effort, the error rate and rejection rate could be traded for one another and could be made quite small. The economical implementation which can be obtained by using analog techniques is described in the next section.

TABLE I

SUMMARY OF MAJOR RECOGNITION RESULTS

*Hand Printing (Alphanumerics)**	
Per Cent Recognition, Alphanumerics	77.2
Per Cent Recognition, Numbers	83.0
Machine Printing (Numerics)†	
Per Cent Recognition	99.7
Per Cent Rejection for no Undetected Errors	0.7

* Results based on the same data used to determine the probability matrices (1800 characters total).

† Results based partly on data used to determine the probability matrices, and partly on additional data (1000 characters total).

An Optical Implementation

Optical Correlation

The character recognition method described in this paper can be economically implemented by electro-optical techniques. The implementation consists essentially of a transparency-photomultiplier combination for each character. The transparency represents the average character. When the image of the unknown character is focused upon the transparency, the transmitted light, measured by the photomultiplier, is a function of the desired cross-correlation value. (This is similar to comparison techniques described by Davis and Norwine[5] and by Bozeman.[6])

Consider a piece of film in which the transparency at each point is proportional to the probability of occurrence of a mark at that point for a particular character. That is, the transparency of the film represents an average character as previously described. Let some input character be focused upon the transparency (Fig. 17). Then the light transmitted through the transparency at a point is a function of the product of the reflectance of the paper and the transmittance of the film at that point. That is, let

$a(x, y)$ = absorption distribution of input pattern,
$r(x, y)$ = reflection distribution of input pattern $= 1 - a(x, y)$,
$t(x, y)$ = transmission distribution of film,
$i(x, y)$ = light intensity transmitted through film.

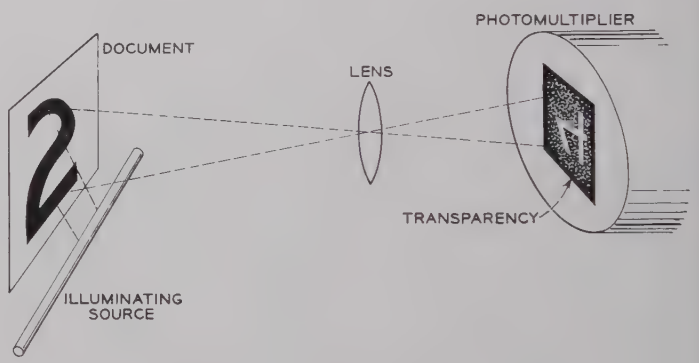


Fig. 17—An optical correlation channel.

Then,

$$i(x, y) \sim r(x, y)t(x, y). \tag{9}$$

If the input pattern is shifted an amount $x = \sigma$, $y = \rho$ with respect to the film then the total light flux, $I(\sigma, \rho)$, transmitted through the film is

$$I(\sigma, \rho) \sim \int\int r(x + \sigma, y + \rho)t(x, y)dxdy \tag{10}$$

or

$$I(\sigma, \rho) \sim \int\int t(x, y)dxdy$$

[5] K. H. Davis and A. C. Norwine, U. S. Patent No. 2,646,465; July 21, 1953.

[6] J. W. Bozeman, U. S. Patent No. 2,898,576; August 4, 1959.

$$-\int\int a(x+\sigma, y+\rho)t(x, y)dxdy, \quad (9)$$

$$I(\sigma, \rho) \sim T - \Phi'(\sigma, \rho), \quad (10)$$

where

T = a constant, different for each character (actually the sum of the probabilities of the probability matrix).

$\Phi'(\sigma, \rho)$ = the modified cross-correlation function between $a(x, y)$ and $t(x, y)$ as a function of the two-dimensional shift σ, p, analogous to (5).

Since $a(x, y)$ represents the mark distribution of the input pattern and $t(x, y)$ represents a probability matrix, we are interested in $\Phi'(\sigma, \rho)$, the cross-correlation function between the input pattern and a given probability matrix.

$I(\sigma, \rho)$ can be measured by a photomultiplier (Fig. 17) which views the entire field of the film. Subtracting T from the output of the photomultiplier will then cause the output to be proportional to the cross-correlation function Φ. This adjustment is easily made by placing a white piece of paper in the field of view. Then $I \sim T$, $a(x, y)$ being arbitrarily taken as zero for white paper. The compensating voltage is then adjusted to make the photocell output zero, making $I \sim \Phi'$ thereafter.

Another normalization is required. It is important that the probability matrices be normalized to some common value, as discussed previously, such that

$$\int\int t^2(x, y)dxdy = N.$$

This normalization is made (once the previous compensation for T has been made) by adjusting the gain of the photochannel.

Penalty Areas

One problem which appears is that of handling penalty areas. Penalty areas are regions of low probability in the probability matrix to which negative weights are assigned. Obviously, one cannot obtain a negative transmittance with a piece of film.

However, note that a constant can be added to every element of every probability matrix in the system. If C is the value of this constant, and P the number of marked elements in the input pattern, then this modification simply causes a constant (PC) to be added to every cross-correlation value. The ordering of the cross-correlation values is not affected and the recognition is still valid.

Therefore, a positive C can be chosen so that its magnitude is equal to that of the greatest penalty weight. The elements of all probability matrices are then assured to be positive after the addition of C.

A Film Correlator

In Fig. 18 a transparency which might be used in an optical correlator is shown. It represents the probability matrix for a machine-printed "2" (from the IBM 704 line printer) with penalty weights in areas of zero probability.

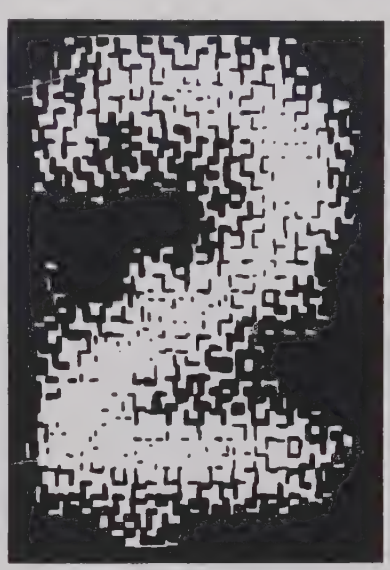


Fig. 18—A transparency for an optical correlator.

It was constructed by extending a 12×12 probability matrix (determined by the IBM 704 computer) to a 36×36 matrix by interpolation. A constant was added to each element so that the penalty areas all had zero weight. Then each element in the matrix was filled in with ink so that the proportion of area left unfilled was the ratio of the weight of that element to the largest weight in the matrix. Hence, unit probability elements are completely open, whereas zero probability elements are filled in completely. Note that, although this example indicates quantization, the quantization can be made arbitrarily small at the expense of additional computer time.

A Recognition System

The basic components of a character recognition system using optical correlation are shown in Fig. 19 for the case of four channels. The extension to n channels is obvious. The combination of the document motion and rotating mirror creates the required two-dimensional shift of the input pattern with respect to each probability matrix. The optical correlator has been discussed above. Each feeds an analog storage device, which, in turn, drives the comparator. The analog storage consists simply of a diode fed capacitor. A possible basic form of the comparator is shown in Fig. 19; briefly only that transistor with the largest base voltage will be conducting when gated. Note that the

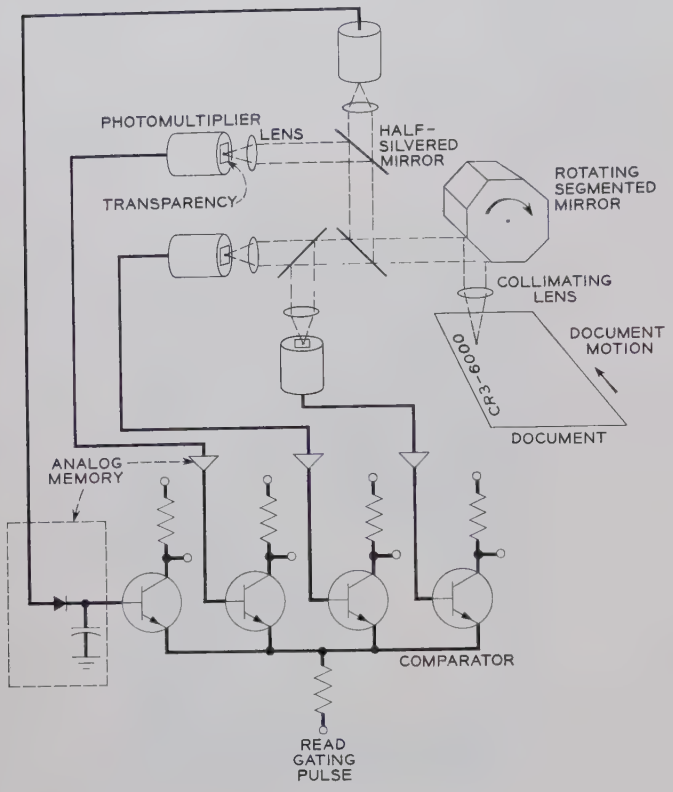


Fig. 19—A character-recognition system using optical correlation.

optics are arranged so that each light path goes through the same number of half-silvered mirrors and lenses to equalize loss, although this is not a fundamental requirement. Additional circuitry is of course required for the various control functions, such as timing, rejection decisions, and resetting the analog store.

Conclusion

A character recognition method capable of an economical analog implementation using optical techniques has been proposed. This method has been simulated on the IBM 704 computer and has been shown to be applicable to machine printing and perhaps to constrained hand printing. The author feels that this recognition method exemplifies some of the many advantages (such as low cost and lack of quantizing error) that can be gained by considering analog implementation in the construction of recognition and allied equipment.

Acknowledgment

The author would like to thank N. Bronstein and J. W. Hreshko for their efforts in the scanning of the characters, and Mrs. R. T. Matsumato for her work in reducing the data for the rejection studies. He would also like to thank W. C. Ridgway for his help in the initial stages of the programming.

The Hall-Effect Analog Multiplier*

G. KOVATCH†, student member, ire, and W. E. MESERVE†, senior member, ire

Summary—**The application of the Hall effect to a general-purpose four-quadrant multiplier is discussed. Circuit diagrams for the transistor amplifiers are given. An evaluation of the experimental results is given for a breadboard model of the multiplier. Static accuracies on the order of 1 per cent to 3 per cent are obtained for the Hall channel and the magnetic channel, respectively. Bandwidths of 25 kc and 1.3 kc are achieved for the Hall channel and the magnetic channel, respectively.**

Introduction

THIS PAPER discusses an analog multiplier which was constructed using an indium arsenide Hall-effect element as the basic multiplying device. As is widely known today, a direct means of analog multiplication is obtained by subjecting the charge carriers in a current-carrying semiconductor or conductor to the action of a magnetic field.[1–8] The voltage which is developed in the material (the Hall voltage) is in a direction mutually perpendicular to the

* Received by the PGEC, February 16, 1961.
† Cornell University, Ithaca, N. Y.

[1] W. Shockley, "Electrons and Holes in Semiconductors," D. Van Nostrand Co., Inc., Princeton, N. J.; 1950.
[2] O. Lindberg, "Hall effect," Proc IRE, vol. 40, pp. 1414–1419; November, 1952.
[3] I. M. Ross, E. W. Saker, and N. A. C. Thompson, "The Hall effect compass," *J. Sci. Instr.* vol. 34, pp. 479–484; December, 1957.
[4] L. Lofgren, "Analog multiplier based on the Hall effect," *J. Appl. Phys.*, vol. 29, pp. 158–166; February, 1958.
[5] R. P. Chasmar and E. Cohen, "An electrical multiplier utilizing the Hall effect in indium arsenide," *Electronic Engr.*, vol. 30, pp. 661–664; November, 1958.
[6] M. J. O. Strutt, "Hall effect in semiconductor compounds," *Electronic and Radio Engr.*, vol. 36, pp. 2–10; January, 1959.
[7] N. P. Milligan, "The Magnetic Circuit, Key to Successful Applications of the Hall Effect," presented at Special Conf. on Nonlinear Magnetics, Washington, D. C.; September 23–26, 1959.
[8] G. Kovatch, "The Hall Effect and Its Application to Multiplying Devices," M.S. thesis, Cornell University, Ithaca, N. Y.; February, 1960.

current and the magnetic flux, and is given by

$$V_H = \frac{R_H i_c B_z}{t},$$

where

V_H = Hall voltage
R_H = Hall constant
i_c = control current
B_z = magnetic flux density
t = thickness of material.

The relative directions of these parameters in the material are shown in Fig. 1.

By suitably arranging the Hall element in a magnetic field such that the current through the element is proportional to one variable to be multiplied and the flux density is proportional to the other variable, a simple means of direct analog multiplication is obtained; the Hall voltage being proportional to the product of the two variables. This is accomplished by placing the Hall element in the airgap of a magnetic iron core such as the one shown in Fig. 2.

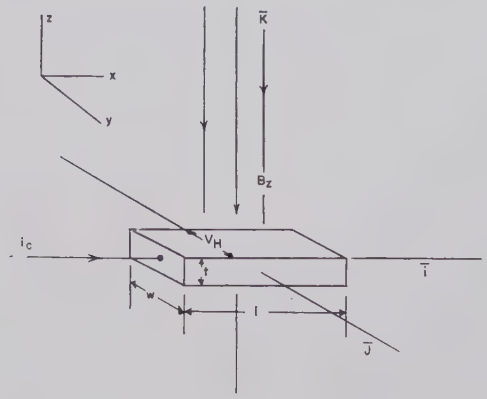


Fig. 1—Hall-effect parameters in a typical Hall element.

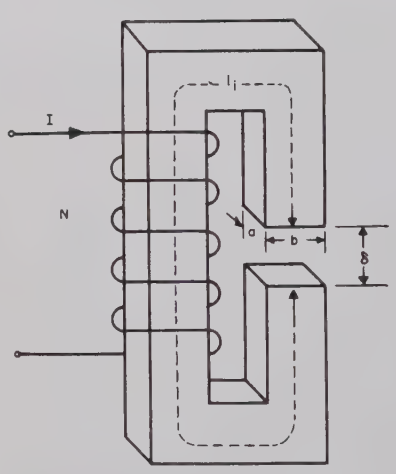


Fig. 2—Magnetic circuit.

Hall Analog Multiplier

Transistor amplifiers were used in the multiplier throughout to provide the signal conversion and amplification needed to produce the control current and the magnetic field current of large enough amplitude. The complete circuit diagram for the multiplier is shown in Fig. 3.

Input Amplifier

Amplifiers A_1 and A_2 are identical since both the Hall element and the magnetic circuit are designed for maximum currents of about one-half ampere. The amplifier is connected in complementary symmetry to provide push-pull dc operation and zero output for zero input conditions.[9] Biasing is essentially class AB with the quiescent condition as low as possible without causing crossover distortion. This achieves the lowest standby power requirement. *P-N-P* power transistors are used with an appropriate *n-p-n* or *p-n-p* driving transistor in the Darlington connection to obtain high current gain and the effect of an *n-p-n* power transistor. The 10-ohm resistors in the emitters of the power combination transistors and the 33-ohm resistors from base to emitter of the power transistor provide some degree of drift stabilization. To achieve a higher degree of drift stabilization, techniques such as chopper stabilization would be required. However, for purposes of this investigation the simpler method was found to be adequate. Both the Hall element and the magnetic circuit are coupled to the input amplifier with 20-ohm resistors. This gives the effect of a current source at the

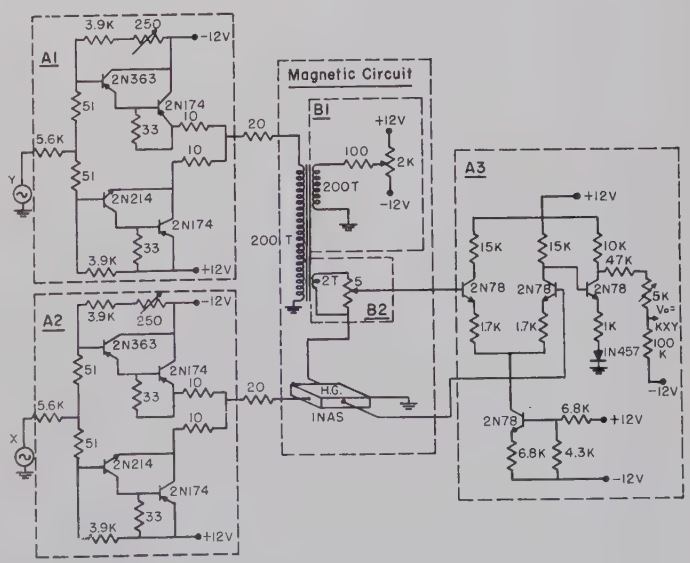


Fig. 3—Schematic diagram of Hall multiplier.

[9] D. L. Greer, "High Power Transistor Amplifiers," General Electric Co., Ithaca, N. Y., Rept. No. TIS-R 58ELC9; January, 1958.

low-impedance levels of the magnetic circuit in the frequency range of interest. Thus, the coil current is proportional to the input voltage for frequencies within the bandwidth of the coil. This resistor also tends to linearize the operation of the Hall element by swamping out the change in input impedance of the element caused by the changing magnetic flux. The input impedance of the Hall element is nominally about one ohm.

Output Amplifier

As the Hall element is a four-terminal device, only one terminal may be grounded at a time. The necessary isolation is obtained by using a difference amplifier as the first stage of the product amplifier A_3. An additional transistor is used in the common emitters of the difference amplifier to provide better isolation by acting as a constant-current generator. Additional stages of dc amplification must be used following the difference amplifier to bring the output voltage to a desired level. Here only one stage is shown, as further amplification is achieved by using a normal operational amplifier.

Compensating Adjustments

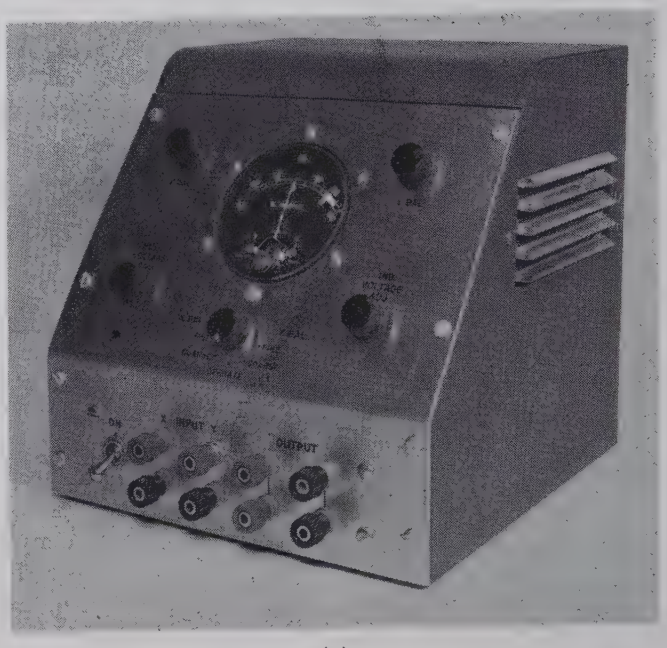
There are two inherent error voltages in the Hall element. The first, the "null voltage," is caused by the misalignment in attaching the output leads along an equipotential line. A small voltage proportional to the control current is superimposed on the output voltage. This voltage is minimized by supplying a dc flux to the airgap of the magnetic circuit as shown in B_1 of Fig. 3. The flux density sets up a voltage proportional to the control current which can be adjusted to be equal in magnitude and opposite in polarity to the null voltage.

The other error voltage is induced in the small loop formed by bringing out the Hall output leads. This voltage is proportional to the rate of change of flux in the airgap. It is compensated for by adding to the output voltage a small voltage which is equal in magnitude and opposite in phase. This compensating voltage is a two-turn winding on the core as shown in B_2 of Fig. 3.

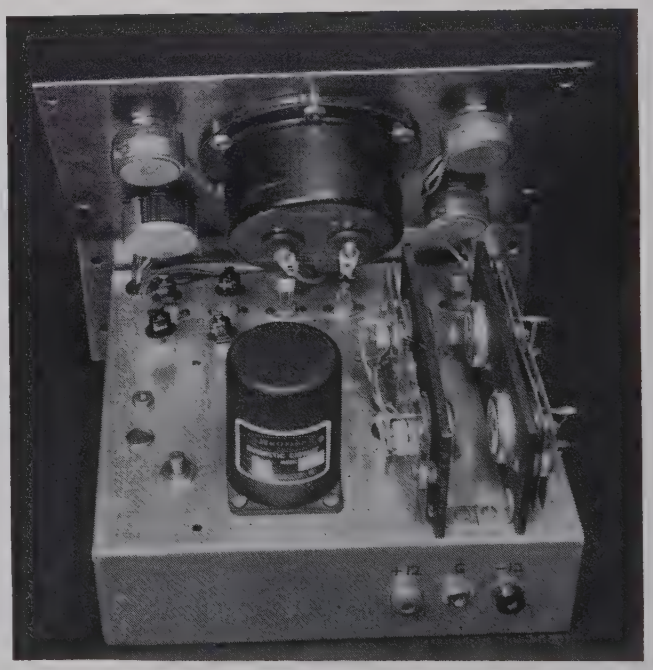
Experimental Results

From the circuit diagram of Fig. 3 a workable model was constructed, as shown in Fig. 4, using conveniently-available components and transistors. The Hall magnetic circuits used were manufactured by the Ohio Semiconductors Company using an indium arsenide Hall element potted into a silectron magnetic iron core. Two units which saturated with 200 ampere-turns were tested. One had 200 turns and the other, 400 turns. As the inductance of the first was one-fourth of the second, its frequency response was four times greater, whereas its flux density was only one-half for the same input conditions. Thus, for the same input power and the same core geometry there was a trade-off between bandwidth and output.

As the frequency response of the Hall element extended into the megacycle range, the bandwidth limitation in the Hall channel was due to the driving amplifier. The results of the frequency response tests are given in Fig. 5. It is seen that the Hall channel is flat to 10 kc, with a bandwidth of 25 kc and a phase shift of 18° at 10 kc. On the other hand, the magnetic channel is limited to a bandwidth of 1.3 kc, being flat to 400 cps, and having 15° phase shift at 400 cps (for the coil with 200 turns). Figs. 6 and 7 demonstrate the step response of the multiplier. Fig. 6 shows the response of the Hall channel to a 1-kc square wave, and Fig. 7 shows the step response of the magnetic channel to a 200-cps square wave.



(a)



(b)

Fig. 4—The Hall-effect analog multiplier. (a) Front view. (b) Rear view.

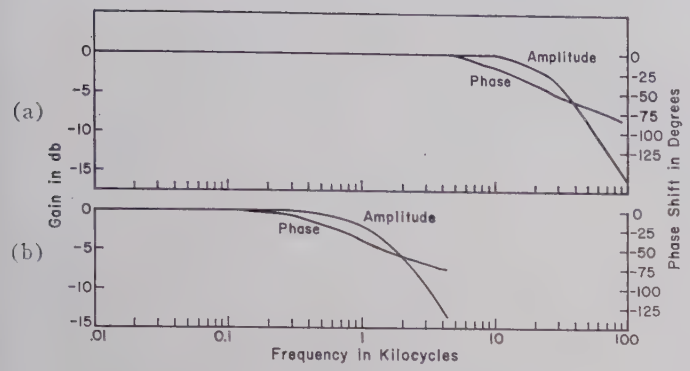


Fig. 5—(a) Frequency response of Hall channel. (b) Frequency response of magnetic channel.

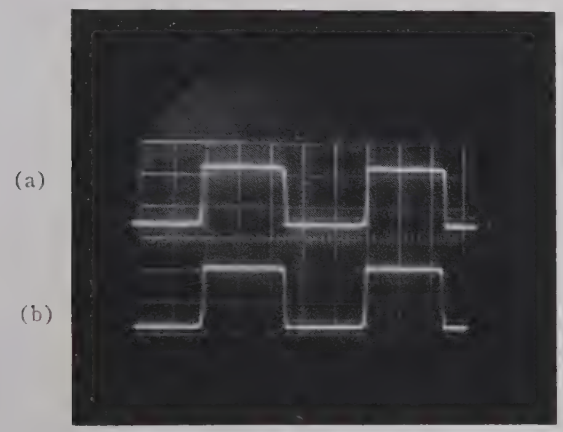


Fig. 6—Hall channel step response for 1-kc square wave. (a) input. (b) output.

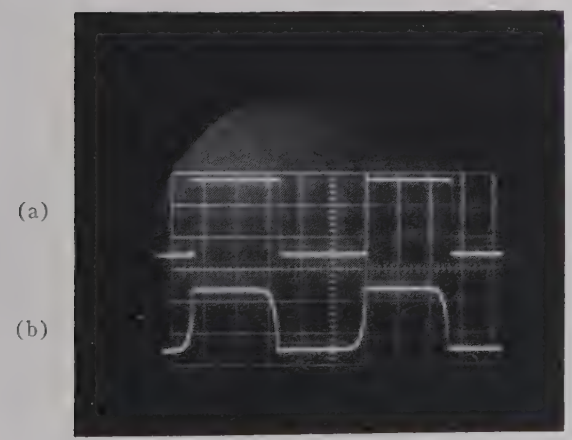


Fig. 7—Magnetic channel step response for 200-cps square wave. (a) Input. (b) Output.

The static or dc error was evaluated by varying the input of one channel from +50 volts through −50 volts (and back to +50 volts for the magnetic channel to show the effect of hysteresis). The other channel had a fixed input of either +50 volts or −50 volts dc. Four-quadrant multiplication was demonstrated in this manner. Fig. 8(a) shows the static error of the Hall channel. It is seen that the error is within ±0.8 per cent full scale. Fig. 8(b) demonstrates the effect of magnetic hysteresis in the magnetic circuit. The total error shown includes the static error of the amplifiers plus the

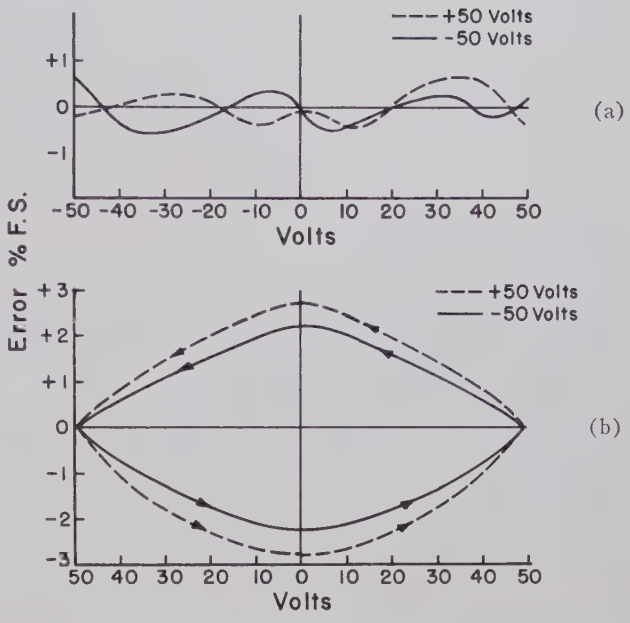


Fig. 8—(a) Static error Hall channel. (b) Total error magnetic channel.

hysteresis error. It is seen that the error for this particular magnetic circuit is about ±2.8 per cent full scale over the range of operation.

Conclusions

Of the many methods for analog multiplication, application of the Hall effect gives perhaps the most direct method for four-quadrant multiplication. It provides relatively wide frequency response and fair accuracy. Some multipliers which have a high degree of accuracy are quite limited in bandwidth, for example, the servomultiplier. Higher-frequency multipliers, such as electronic time division multipliers, have a higher error and also are much more expensive.

The Hall multiplier discussed in this paper is not viewed in its present form as optimum nor as the best that can be done with this scheme. Tests and evaluations were not exhaustive, and many improvements in circuit design and other investigations of core materials and Hall elements can be made. The major aim here was to demonstrate the application of the Hall effect in a workable general-purpose four-quadrant multiplier. With more sophisticated circuit design and with better magnetic core materials, much better frequency response and accuracy may be expected. In this way the Hall effect should find much wider application in analog multiplication.

Acknowledgment

The authors wish to express appreciation to the following institutions and persons for their assistance in the performance of this research: Cornell University for provision of facilities and materials, General Electric Company's Advanced Electronics Center and, in particular, David L. Greer and the other engineers who provided expert assistance.

Copper-Mandrel Potentiometer Dynamic Error and Compensation*

C. H. SINGLE†, MEMBER, IRE, AND J. A. BRUSSOLO†

Summary—A simple potentiometer equivalent circuit is presented that is valid for single or multiturn, copper-mandrel, wirewound precision potentiometers. Developed to obtain a practical approximation of potentiometer ac characteristics, it is particularly useful in error analyses where small phase errors are critical.

Capacitive compensation techniques are also given that can achieve considerable reduction in potentiometer dynamic error. Even for heavy capacitive loading it is possible virtually to eliminate potentiometer phase error (important in most analog computer circuits). Correspondingly, high-frequency square waves can be attenuated reasonably by such capacitively-compensated potentiometers.

I. Introduction

A CONSIDERABLE amount of work has been done in the analysis of copper-mandrel potentiometer ac characteristics.[1,2] This paper examines the suggested equivalent circuit using simple laboratory techniques to confirm its validity. The equivalent circuit is then used to develop universal low-frequency phase-error curves for various potentiometer capacitive loads. Next, it forms the basis for calculating phase-error reduction through capactive-compensation techniques. The benefits of such capacitive compensation are confirmed experimentally for a ten-segment pot for both low-frequency phase error and high-frequency transient response.

An uncompensated 50-K potentiometer with only 300 pf of capacitive load will have, at 100 cps, a phase error of approximately 0.39° lead for zero displacement and a maximum lag of 0.16° for 0.648 displacement. This small phase error corresponds to maximum amplitude errors of 0.68 per cent and 0.29 per cent, respectively. In a large-scale analog computer, it is difficult to hold potentiometer arm to ground capacitance below 300 pf since capacitive loading includes potentiometer arm to patchboard wiring, patchcord interconnection, patchboard to load wiring, and the capacitive component of the load. If signal components exceed 100 cps, potentiometer phase error can be a significant error source in an analog computer.

II. Confirmation of the Equivalent Circuit

An oscilloscope can conveniently be used to detect zero phase shift as shown in Fig. 1. The initial objective was primarily to determine the value of C_0 and C_1 (potentiometer equivalent capacity), assumed initially to be a function of potentiometer displacement, and subsequently to use these values in an equivalent circuit. The potentiometer and oscilloscope circuit was capacitively compensated as required by either C_x or C_y at many potentiometer displacements to achieve zero phase shift as indicated by the standard linear oscilloscope Lissajous pattern. The main precaution necessary for accurate phase nulling is to cancel the oscilloscope horizontal vs vertical amplifier phase discrepancies at each frequency. It is also helpful to have balanced horizontal and vertical scope amplifiers of high sensitivity to obtain accurate data.

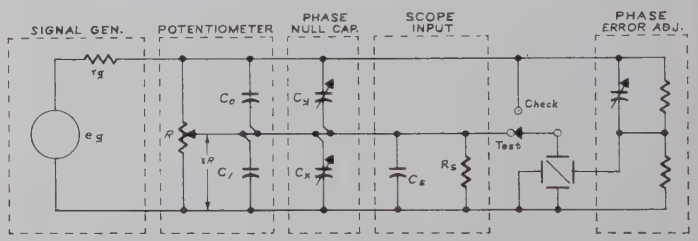


Fig. 1—Potentiometer test circuit.

Type-A Helipots® proved to be symmetrical, that is, C_0 equal to C_1. This was confirmed on potentiometers having various resistances (10K, 20K, 30K, 50K, and 100K) by clockwise and counterclockwise terminal reversal. Excellent agreement between C_x and C_y was observed for values of one displacement as compared with its complement. C_0 is determined by

$$C_0 = \frac{\gamma(C_s + C_x) - (1 - \gamma)C_y}{1 - 2\gamma}. \qquad (1)$$

For other potentiometers, construction details may cause slight differences in C_0 and C_1. This is not important since the C_0 and C_1 values can similarly be determined. Any difference in C_1 can be absorbed as part of the external capacitive load C_e, and thus the curves later developed have general application.

With typical potentiometer data, as shown in Fig. 2, C_0 surprisingly proved to be essentially invariant for *all values* of potentiometer displacement ($0.05 < \gamma < 0.95$). From the above information, the simple equivalent circuit, Fig. 3, was established.

Values for C_0 for the type-A Helipots® differ slightly with potentiometer size as shown in Table I.

* Received by the PGEC, August 22, 1960; revised manuscript received, March 23, 1961.

† Berkeley Div., Beckman Instruments, Inc., Richmond, Calif.

[1] C. H. Single, "Equivalent Circuit for Copper-Mandrel Potentiometers," Berkeley Div., Beckman Instruments, Inc., Richmond, Calif., 15 pp.; June, 1959.

[2] B. F. Logan, "AC Performance and Phase Compensation of Copper-Mandrel Potentiometers," Helipot Div., Beckman Instruments, Inc., Fullerton, Calif., Tech. Paper No. 497; January, 1955.

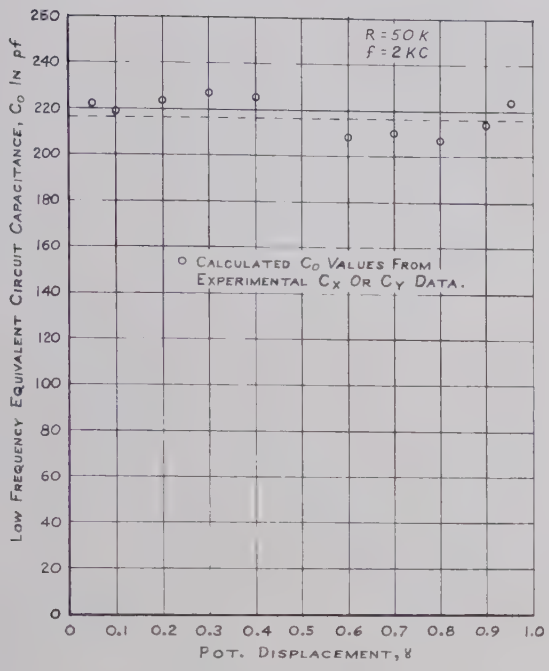


Fig. 2—Potentiometer capacitance vs potentiometer displacement.

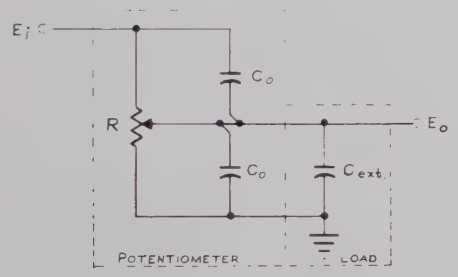


Fig. 3—Equivalent circuit.

TABLE I

R	C_0	Pot to Mandrel Capacitance, $C \approx 12C_0$	$f_{max}^* = \frac{0.1592}{RC}$
10 K	200 pf	2400 pf	6.6 kc
20 K	204 pf	2450 pf	3.3 kc
30 K	208 pf	2500 pf	2.1 kc
50 K	216 pf	2600 pf	1.2 kc
100 K	224 pf	2700 pf	590 cps

* Based on the $f < \omega RC$ limit suggested in the references.[1-3]

The transfer characteristic of the equivalent circuit is

$$E_0(s) = \frac{\gamma[(1-\gamma)RC_0s + 1]E_i(s)}{\gamma(1-\gamma)R(C_0 + C_1)s + 1} = \frac{\gamma(\tau_1 s + 1)E_i(s)}{(\tau_2 s + 1)}, \quad (2)$$

where

$$C_1 = C_0 + C_{ext}.$$

Eq. (2) is a simple lead-lag, or lag-lead transfer characteristic, easy to take into account when performing error analysis. Phase shift for a sinusoidal input voltage is given by

$$\phi = \arctan \omega\tau_1 - \arctan \omega\tau_2. \quad (3)$$

However, in most analog-computer error analysis the frequencies involved are low enough to allow even further simplification. Low-frequency phase shift can be adequately approximated by using the net time constant, $\tau_1 - \tau_2$, of (2) and (3).

$$\phi \approx \omega\tau_{net} = \omega(\tau_1 - \tau_2) = \omega RC_0(1-\gamma)(1 - 2\gamma - k\gamma), \quad (4)$$

where $-\phi$ = a lagging phase angle, $k = C_{ext}/C_0$, $\omega\tau_1 < 0.1$, and $\omega\tau_2 < 0.1$.

Universal curves showing normalized phase shift, $\phi/\omega RC_0$, are given in Fig. 4. These curves are valid for any potentiometer value, capactive loading, and potentiometer displacement. They apply in general to any single turn or multi-turn copper-mandrel potentiometer. Values of C_0 for other potentiometers can be determined experimentally or by

$$C_0 \approx \frac{C}{12} \quad (5)$$

if the mandrel to winding capacitance C is known.[1,2]

The amplitude error at low frequencies can be obtained from (2):

$$\epsilon = \gamma - \gamma\left[\frac{1 + \tau_1^2\omega^2}{1 + \tau_2^2\omega^2}\right]^{1/2} \quad (6)$$

$$\approx \tfrac{1}{2}\gamma(\tau_2^2 - \tau_1^2)\omega^2. \quad (7)$$

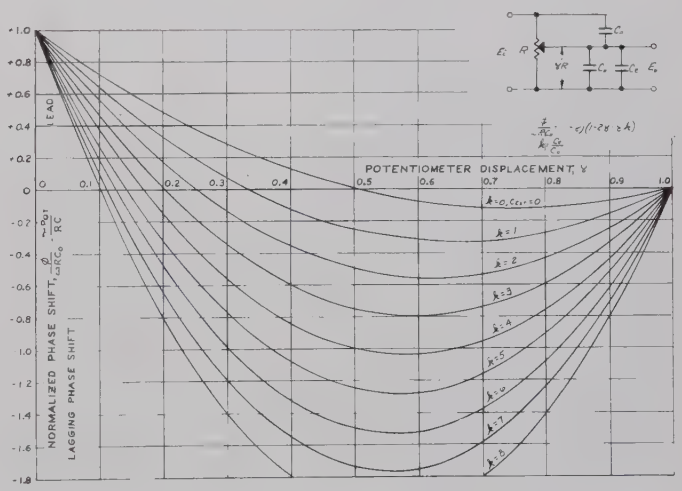


Fig. 4—Potentiometer normalized phase shift vs displacement.

Comparing (4) and (7), it can be seen that the amplitude error is smaller than the phase error by an order of magnitude in the frequency variable ω. Thus, the potentiometer error can be adequately represented by considering the phase-shift error only.

III. Capacitive-Compensation Techniques

In a large-scale analog computer, many potentiometers are used for scaling to achieve the various problem coefficients. Due to practical physical limitations, the

potentiometers are located somewhat remotely from their loads. For flexibility in computer component arrangement, this generally means the potentiometer is wired to a patchboard for operator assignment. Usually, one unshielded wire is brought to the patchboard from the top of the potentiometer for the low-impedance driving voltage, one shielded wire is brought to the patchboard from the arm of the potentiometer for connection to its assigned load and, finally, the bottom of the potentiometer is grounded. The shielding of the potentiometer arms is necessary to avoid cross-coupling of undesired driving voltages to other potentiometer arms through the capacitive coupling of potentiometer wiring cables.

With operator load selection, the practical potentiometer circuit is that of Fig. 5, where R_{ext} is usually a single resistive element, typically the dc portion of a following amplifier input impedance. C_{ext} is the combined capacitance from potentiometer arm shielding, patchcord wiring capacitance, shielded wire loading resistor capacitance, and load-resistance compensation capacitance (if used).

For any particular potentiometer displacement γ, it would be possible to add a capacitor from the arm of the potentiometer to either the top or bottom of the potentiometer to cancel the phase error perfectly. This is true for any value of loading as long as R_{ext} and C_{ext} are fixed for that particular displacement. If R_{ext} and C_{ext} are functions of time or some other parameter, the compensation would be much more complex. Compensation for only fixed (but arbitrary) potentiometer loading is developed here.

Being able to compensate each potentiometer for any fixed resistive and capacitive load by such a simple means as adding one capacitor is helpful. However, it is not very practical, since the compensating capacitor varies with both load and displacement. Further, the capacitor location will vary with displacement. Finally, there are too many potentiometers involved to afford this awkward technique.

Various means of potentiometer compensation are suggested by Schneider, *et al.*[3] Most are suitable for compensating only the potentiometer error, not the loaded potentiometer error. However, the type "*f*" compensation discussed in Schneider[3] readily allows extension to include arbitrary pot-loading effects. The technique is to place capacitors from a few distributed potentiometer taps at either the top or bottom of the pot, to correctly compensate phase error for pot displacement corresponding to tap location.

The circuit analysis of a pot under load, with only a few taps, is quite involved. However, this has been carried out for several cases to show the nature of the

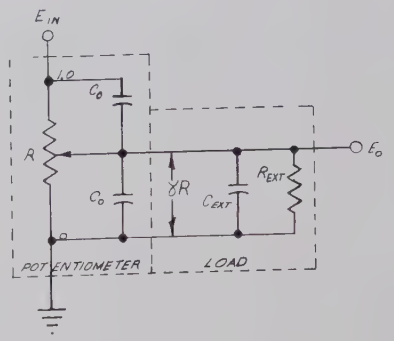


Fig. 5—Computer potentiometer equivalent circuit.

phase-error curves. Universal curves have been plotted to enable an operator to select compensating capacitors for optimum phase-shift characteristics.

The simplest capacitive compensation that can be used is to place one tap on the pot, at the midpoint. A compensating capacitor can then be used between the pot tap and the pot input. The normalized phase-shift curves for this case are shown in Fig. 6, where the compensating capacitor has been taken equal to the external capacitor. These should be compared with the phase-shift curves for a pot with no taps, as shown in Fig. 4. It can be seen that the phase shift for the tapped pot is reduced considerably compared to that for the untapped case, for pot displacements greater than 0.2. The normalized phase shift for the one-tap case is given by

$$\frac{\phi}{\omega RC_0} = (1-\gamma)(1-2\gamma-\gamma k) + \begin{cases} \frac{1}{4}\alpha k & 0 \le \gamma \le \frac{1}{2} \\ \dfrac{1-\gamma}{4\gamma}\alpha k & \frac{1}{2} \le \gamma \le 1, \end{cases} \tag{8}$$

where $k = C_e/C_0$.

A greater improvement can be obtained by considering a pot with two equally-spaced taps and two compensating capacitors. The normalized phase shift for the two-tap case is given by

$$\frac{\phi}{\omega RC_0} = (1-\gamma)(1-2\gamma-\gamma k) + \begin{cases} \dfrac{1}{9}(\alpha-2\beta)k & 0 \le \gamma \le \frac{1}{3} \\ \dfrac{1}{9}\left[\alpha - \left(\dfrac{1-\gamma}{\gamma}\right)\beta\right]k & \frac{1}{3} \le \gamma \le \frac{2}{3} \\ \dfrac{1-\gamma}{9\gamma}(2\alpha-\beta)k & \frac{2}{3} \le \gamma \le 1 \end{cases} \tag{9}$$

where $k = C_e/C_0$, and α and β are factors representing the capacitors used on the pot taps.

In order to appreciate the significance of the two-tap pot, (9) has been solved for the values of α and β necessary to make the phase shift zero at the two taps. This gives the normalized phase-shift curves shown in Fig. 7.

[3] S. Schneider, F. Hiroaka, and C. Gauldin, "Measurement and Correction of Phase Shift in Copper-Mandrel Precision Potentiometers," Helipot Div., Beckman Instruments, Inc., Fullerton, Calif., Tech. Paper No. 552; June, 1957.

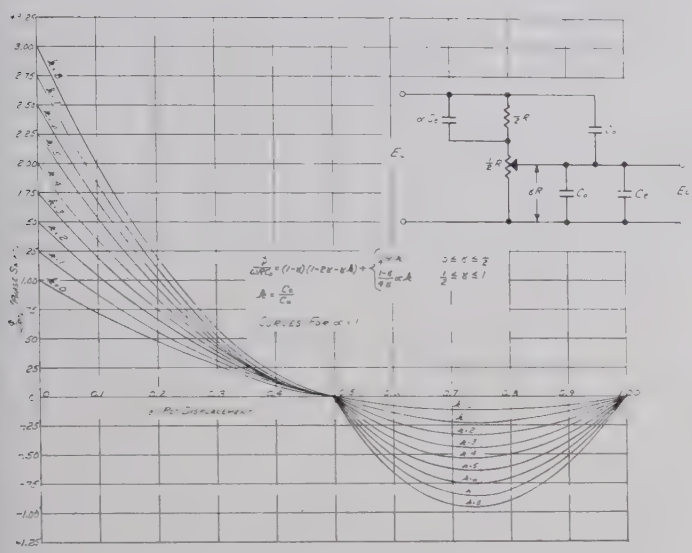


Fig. 6—Potentiometer normalized phase shift vs displacement (one tap).

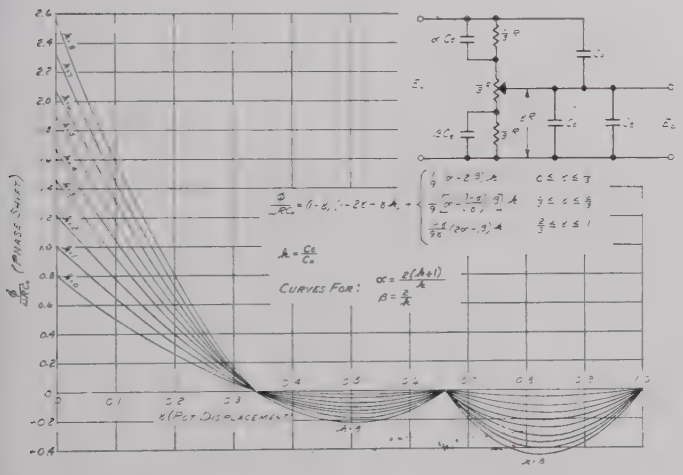


Fig. 7—Potentiometer normalized phase shift vs displacement (two taps).

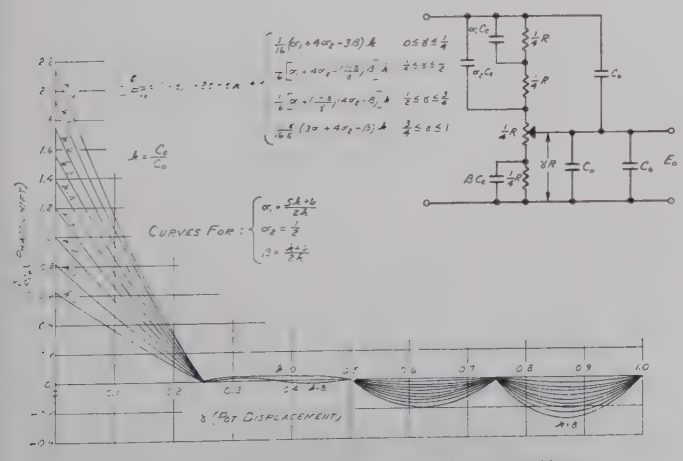


Fig. 8—Potentiometer normalized phase shift vs displacement (three taps).

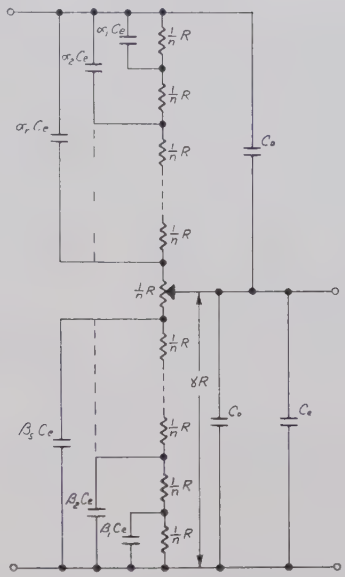


Fig. 9—Circuit configuration for m-taps. m=number of taps, n=number of pot segments$=m+1$, γ=pot displacement, R=total pot resistance, C_0=equivalent end-to-arm pot capacitance, C_e=external capacitance load, r=number of capacitors going up$=n-s-1$, s=number of capacitors going down$=n-r-1$, $k=C_e/C_0$.

Again, it should be noticed that the phase error has been decreased by adding a tap.

Curves are shown in Fig. 8 for a pot with three taps. As was done for the previous cases, the values of the tap capacitors have been determined in order to make the phase shift equal to zero at the taps.

By looking at Figs. 4 and 6–8, three characteristics of adding capacitors and taps to a pot can be deduced. First, it can be seen that the phase shift for the first segment of the pot tends to be relatively large and is a leading phase shift. Second, the phase shift for succeeding sections is relatively small but gradually increases in amplitude. Third, there is a crossover point between a leading phase shift and a lagging phase shift, as evidenced in Fig. 8 for $\frac{1}{4}\leq\gamma\leq\frac{1}{2}$. It can be shown (by considering pots with more taps) that this phase-shift crossover point occurs between taps where the last tap capacitors go down and up, respectively. In other words, it is simple to select a region of pot displacement for minimum phase shift. All tap capacitors for smaller displacements must go down and all tap capacitors for larger displacements must go up.

In general, it is evident that more taps decrease the pot phase shift. Equations and curves could be worked out for many individual cases, but their use would be restricted to specific problems. Therefore, an attempt has been made to work out the form of the phase-shift equations for the general case. Fig. 9 shows the nomenclature involved. The pot is assumed to have equally-spaced taps throughout, and a compensating capacitor on each tap. It is also assumed that at least one capacitor goes up and, for more than one tap, that at least one capacitor goes down.

The phase shift is always composed of two terms: the phase shift for the no-tap pot, and a term which is a function of the tap capacitors and which changes for each pot segment. In general, the phase shift is given by

$$\frac{\phi}{\omega RC_0} = (1-\gamma)(1-2\gamma-\gamma k) + \begin{cases} \frac{k}{n^2} f_1 & 0 \le \gamma \le \frac{1}{n} \\ \frac{k}{n^2} f_2 & \frac{1}{n} \le \gamma \le \frac{2}{n} \\ \frac{k}{n^2} f_3 & \frac{2}{n} \le \gamma \le \frac{3}{n} \\ \vdots & \vdots \\ \frac{k}{n^2} f_{n-1} & \frac{n-2}{n} \le \gamma \le \frac{n-1}{n} \\ \frac{k}{n^2} f_n & \frac{n-1}{n} \le \gamma \le 1. \end{cases} \tag{10}$$

The functions $f_1, f_2 \cdots, f_n$ are functions of the tap capacitors $\alpha_1, \alpha_2, \cdots, \alpha_r$; $\beta_1, \beta_2, \cdots, \beta_s$. They are given by the following equations:

$$f_1 = \alpha_1 + (2)^2\alpha_2 + (3)^2\alpha_3 + \cdots + (r-1)^2\alpha_{r-1} + r^2\alpha_r - [s(n-s)\beta_s + (s-1)(n-s+1)\beta_{s-1} + (s-2)(n-s+2)\beta_{s-2} + \cdots + 3(n-3)\beta_3 + 2(n-2)\beta_2 + (n-1)\beta_1]$$

$$f_2 = \alpha_1 + (2)^2\alpha_2 + \cdots + r^2\alpha_r - \left[s(n-s)\beta_s + \cdots + 3(n-3)\beta_3 + 2(n-2)\beta_2 + \frac{(1-\gamma)}{\gamma}\beta_1\right]$$

$$f_3 = \alpha_1 + (2)^2\alpha_2 + \cdots + r^2\alpha_r - \left\{s(n-s)\beta_s + \cdots + 3(n-3)\beta_3 + \frac{(1-\gamma)}{\gamma}[(2)^2\beta_2 + \beta_1]\right\}$$

$$\vdots$$

$$f_s = \alpha_1 + (2)^2\alpha_2 + \cdots + r^2\alpha_r - \left\{s(n-s)\beta_s + \frac{(1-\gamma)}{\gamma}[(s-1)^2\beta_{s-1} + (s-2)^2\beta_{s-2} + \cdots + (3)^2\beta_3 + (2)^2\beta_2 + \beta_1]\right\}$$

$$f_{s+1} = \alpha_1 + (2)^2\alpha_2 + \cdots + r^2\alpha_r - \frac{(1-\gamma)}{\gamma}\cdot[s^2\beta_s + (s-1)^2\beta_{s-1} + \cdots + (2)^2\beta_2 + \beta_1]$$

$$f_{s+2} = \alpha_1 + (2)^2\alpha_2 + \cdots + (r-2)^2\alpha_{r-2} + (r-1)^2\alpha_{r-1} + \frac{(1-\gamma)}{\gamma} r(n-r)\alpha_r - \frac{(1-\gamma)}{\gamma}[s^2\beta_s + \cdots + (2)^2\beta_2 + \beta_1]$$

$$f_{s+3} = \alpha_1 + (2)^2\alpha_2 + \cdots + (r-2)^2\alpha_{r-2} + \frac{(1-\gamma)}{\gamma}[(r-1)(n-r+1)\alpha_{r-1} + r(n-r)\alpha_r] - \frac{(1-\gamma)}{\gamma}[s^2\beta_s + \cdots + (2)^2\beta_2 + \beta_1]$$

$$\vdots$$

$$f_{n-1} = \alpha_1 + \frac{(1-\gamma)}{\gamma}[2(n-2)\alpha_2 + 3(n-3)\alpha_3 + \cdots + (r-1)(n-r+1)\alpha_{r-1} + r(n-r)\alpha_r] - \frac{(1-\gamma)}{\gamma}[s^2\beta_s + \cdots + (2)^2\beta_2 + \beta_1]$$

$$f_n = \frac{(1-\gamma)}{\gamma}[(n-1)\alpha_1 + 2(n-2)\alpha_2 + 3(n-3)\alpha_3 + \cdots + (r-2)(n-r+2)\alpha_{r-2} + (r-1)(n-r+1)\alpha_{r-1} + r(n-r)\alpha_r] - \frac{(1-\gamma)}{\gamma}[s^2\beta_s + (s-1)^2\beta_{s-1} + \cdots + (3)^2\beta_3 + (2)^2\beta_2 + \beta_1]. \tag{11}$$

These equations can be put in the more compact form given below for the pot segment $i-1/n \le \gamma \le i/n$:

for $0 \le i \le s$:

$$f_i = \sum_{p=1}^{r} p^2\alpha_p - \frac{(1-\gamma)}{\gamma}\sum_{p=1}^{i-1} p^2\beta_p - \sum_{p=1}^{s} p(n-p)\beta_p;$$

for $i = s+1$:

$$f_i = \sum_{p=1}^{r} p^2\alpha_p - \frac{(1-\gamma)}{\gamma}\sum_{p=1}^{s} p^2\beta_p;$$

for $s+2 \le i \le n$:

$$f_i = \sum_{p=1}^{n-1} p^2\alpha_p + \frac{(1-\gamma)}{\gamma}\sum_{p=n-i+1}^{r} p(n-p)\alpha_p - \frac{(1-\gamma)}{\gamma}\sum_{p=1}^{s} p^2\beta_p. \tag{12}$$

A look at the phase-shift curves in Figs. 4 and 6–8 shows that an optimum phase-shift reduction is not obtained through the use of equally-spaced taps. The phase shift over the first pot segment is relatively large. This is generally tolerable in most cases, since few applications require the use of very small coefficients. Also, from an economic standpoint, equally-spaced tapped pots are

easier to manufacture. However, the thought occurs to try to reduce the phase shift with fewer taps through a more judicious choice of tap placement. It is very difficult to find the tap locations for an optimum phase shift, since the equations change their form at each tap. Nevertheless, a good guess can be made about providing reduced phase shift. This has been done for a pot with two taps. The taps were arbitrarily selected to be at $\gamma=0.1$ and $\gamma=0.7$. The equations for the phase shift have been solved to obtain zero phase shift at the tap locations. The normalized phase-shift curves for this case are shown in Fig. 10. Comparing these curves with those in Fig. 8, it can be seen that two taps, properly spaced, given less phase shift than three uniformly-spaced taps.

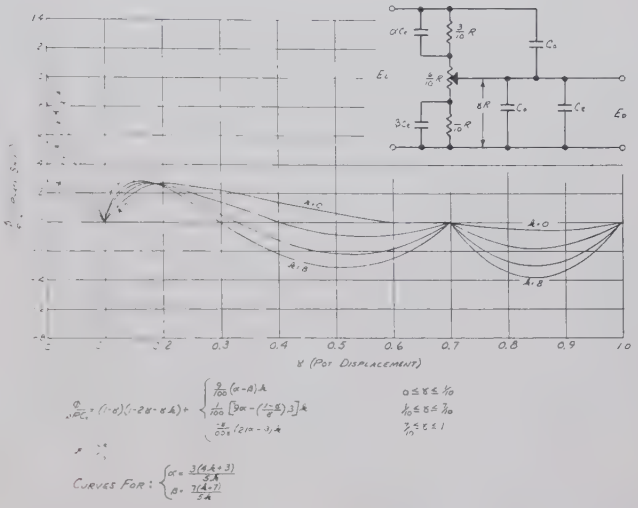


Fig. 10—Potentiometer normalized phase shift vs displacement (two nonuniformly spaced taps).

IV. Experimental Verification

While the analysis developed in Section III can be used to achieve potentiometer compensation, it is also possible to determine the correct tap capacitors by iterative adjustment for minimum phase shift at each tap location. This has been done here for a 30-K type-A Helipot® with 10 resistive segments (nine extra taps) with $R_{ext}=100$ K, and $C_{ext}=940$ pf. This is somewhat larger loading than is normally encountered for potentiometers in analog computers. For less loading, the difference in loading can be added, if one desires to use this particular compensation rather than developing one for the actual case. The measurement technique for phase nulling is that of Single.[1] The recommended circuit is shown in Fig. 11.

Single[1] has shown the value of k to be

$$k=\frac{C_{ext}}{C_0}=\frac{940}{208}\approx 4.5,$$

which results in a large phase error for the uncompensated potentiometer. For

$$k=4.5,\qquad \frac{\phi}{\omega RC_0}\text{ min}=1.17,$$

or $\phi_{min}=1.17\ RC_0\omega$ radians $=2.63(10)^{-3}f$ in degrees. At 100 cps, this would yield a maximum of 0.263° phase lag (at $\gamma\approx 0.58$) from the potentiometer alone. This 0.457 per cent error, due to one potentiometer at 100 cps, does not indicate system or problem error.

The normalized phase error at low frequencies has not been computed. It has been experimentally estimated, as shown in Fig. 12, following the careful selection of tap capacitors shown in Fig. 11. The error curve could be better balanced, particularly for displacements greater than 0.7, by a slight modification of the higher tap capacitors. However, it is quite close and is a considerable improvement over the uncompensated potentiometer.

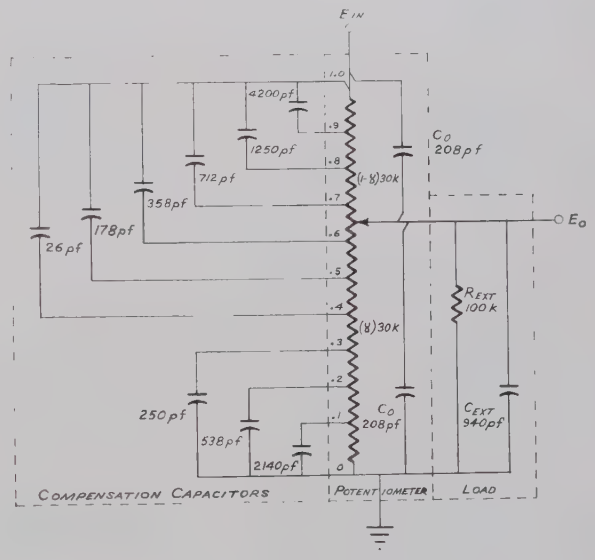


Fig. 11—Experimental circuit.

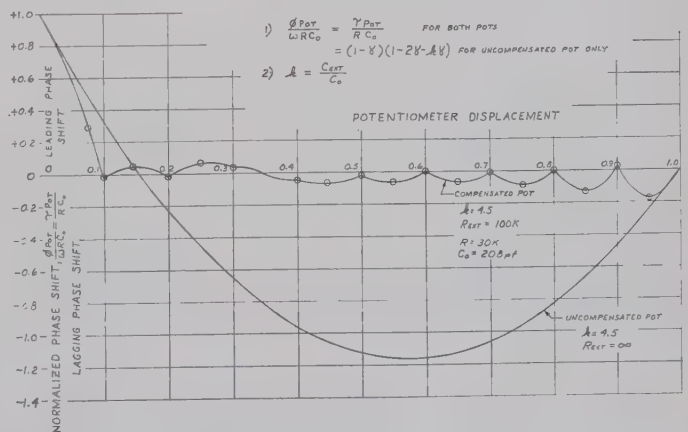


Fig. 12—Potentiometer normalized phase shift vs displacement, showing effect of potentiometer compensation.

To appreciate the resultant improvement in potentiometer high-frequency characteristics, the square-wave response of both compensated and uncompensated potentiometers was checked as given in Fig. 13. Pot displacements of 0.05 increment were used to show both the best possible response ($\gamma = 0.1, 0.2, \cdots, 0.9$) and the poorest response ($\gamma = 0.05, 0.15, 0.25, \cdots, 0.95$). The difference is not readily apparent at the low frequencies (2.5 kc to 25 kc), but is quite evident at the higher frequencies (50 kc to 250 kc).

The uncompensated potentiometer has obvious errors even at 2.5 kc and is quite useless above 25 kc as a square-wave attenuator. The compensated potentiometer is still quite good at 50 kc, *i.e.*, approximately the same error pattern as the uncompensated pot at 2.5 kc.

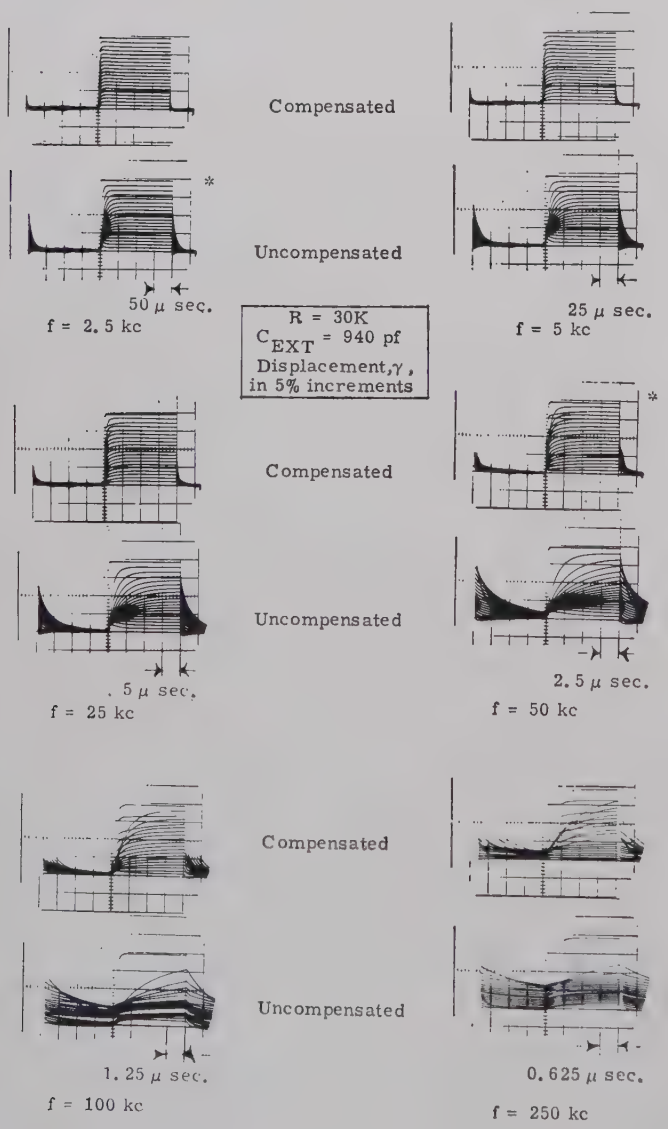


Fig. 13—Oscillograms of square-wave response for copper-mandrel potentiometers, showing effect of capacitive compensation. Asterisk denotes compensated 50-kc response ≈ uncompensated 2.5-kc response.

V. Conclusion

This paper presents a simple equivalent circuit for copper-mandrel potentiometers that is quite useful in low- and moderate-frequency error analyses in analog computers. Equations and graphs are also presented which show that a considerable reduction in potentiometer dynamic error can be achieved through the use of capacitive compensation. Phase-shift curves are given to serve as a guide to compensating potentiometers. Finally, experimental verification is obtained on the advantages of using capacitive compensation with tapped potentiometers. With a few such compensated pots available for critical circuits, the problem of accuracy can be greatly improved.

Appendix

Transfer Functions for Tapped Pots

It may be necessary, at high frequencies, to take into account the phase shift contributed by higher-order terms. For this reason, the entire transfer functions for pots with one and two taps are presented here.

Pot with no taps (see Fig. 4):

$$\frac{E_0(s)}{E_i(s)} = \gamma \frac{(1-\gamma)RC_0 s + 1}{\gamma(1-\gamma)R(2C_0 + C_e)s + 1}.$$

Pot with one tap (see Fig. 6):

$$\frac{E_0(s)}{E_i(s)} = \gamma \frac{As^2 + Bs + 1}{Cs^2 + Ds + 1}$$

for $0 \leq \gamma \leq \frac{1}{2}$

$$A = \tfrac{1}{4}(1-2\gamma)R^2\alpha C_0 C_e$$

$$B = (1-\gamma)RC_0 + \tfrac{1}{2}R\alpha C_e$$

$$C = \tfrac{1}{4}\gamma(1-2\gamma)R^2\alpha C_e(2C_0 + C_e)$$

$$D = \gamma(1-\gamma)R(2C_0 + C_e) + \tfrac{1}{4}R\alpha C_e;$$

for $\frac{1}{2} \leq \gamma \leq 1$

$$A = \frac{1}{4\gamma}(1-\gamma)(2\gamma-1)R^2\alpha C_0 C_e$$

$$B = (1-\gamma)RC_0 + \frac{1}{4\gamma}R\alpha C_e$$

$$C = \tfrac{1}{2}(1-\gamma)(2\gamma-1)R^2\alpha C_0 C_e + \tfrac{1}{4}(1-\gamma)(2\gamma-1)R^2\alpha C_e^2$$

$$D = \gamma(1-\gamma)R(2C_0 + C_e) + \tfrac{1}{4}R\alpha C_e.$$

Pot with two taps (see Fig. 7):

$$\frac{E_0(s)}{E_i(s)} = \gamma \frac{As^3 + Bs^2 + Cs + 1}{Ds^3 + Es^2 + Fs + 1}$$

for $0 \leq \gamma \leq \frac{1}{3}$

$$A = \frac{1}{27}(1 - 3\gamma)R^3\alpha\beta C_0 C_e^2$$

$$B = \frac{1}{9}(2 - 3\gamma)R^2\alpha C_0 C_e + \frac{2}{9}(1 - 3\gamma)R^2\beta C_0 C_e$$

$$C = (1 - \gamma)RC_0 + \frac{1}{3}R\alpha C_e$$

$$D = \frac{1}{27}\gamma(1 - 3\gamma)R^3\alpha\beta C_e^2(2C_0 + C_e)$$

$$E = \frac{1}{9}\gamma(2 - 3\gamma)R^2\alpha C_e(2C_0 + C_e) + \frac{2}{9}\gamma(1 - 3\gamma)R^2\beta C_e(2C_0 + C_e) + \frac{1}{27}R^2\alpha\beta C_e^2$$

$$F = \gamma(1 - \gamma)R(2C_0 + C_e) + \frac{2}{9}R\alpha C_e + \frac{2}{9}R\beta C_e;$$

for $\frac{1}{3} \leq \gamma \leq \frac{2}{3}$

$$A = \frac{1}{81\gamma}(2 - 3\gamma)(3\gamma - 1)R^3\alpha\beta C_0 C_e^2$$

$$B = \frac{1}{9}(2 - 3\gamma)R^2\alpha C_0 C_e + \frac{1}{9\gamma}(1 - \gamma)(3\gamma - 1)R^2\beta C_0 C_e + \frac{1}{27\gamma}(3\gamma - 1)R^2\alpha\beta C_e^2$$

$$C = (1 - \gamma)RC_0 + \frac{1}{3}R\alpha C_e + \frac{1}{9\gamma}(3\gamma - 1)R\beta C_e$$

$$D = \frac{1}{81}(2 - 3\gamma)(3\gamma - 1)R^3\alpha\beta C_e^2(2C_0 + C_e)$$

$$E = \frac{1}{9}\gamma(2 - 3\gamma)R^2\alpha C_e(2C_0 + C_e) + \frac{1}{9}(1 - \gamma)(3\gamma - 1)R^2\beta C_e(2C_0 + C_e) + \frac{1}{27}R^2\alpha\beta C_e^2$$

$$F = \gamma(1 - \gamma)R(2C_0 + C_e) + \frac{2}{9}R\alpha C_e + \frac{2}{9}R\beta C_e;$$

for $\frac{2}{3} \leq \gamma \leq 1$

$$A = \frac{1}{27\gamma}(1 - \gamma)(3\gamma - 2)R^3\alpha\beta C_0 C_e^2$$

$$B = \frac{2}{9\gamma}(1 - \gamma)(3\gamma - 2)R^2\alpha C_0 C_e + \frac{1}{9\gamma}(1 - \gamma)(3\gamma - 1)R^2\beta C_0 C_e + \frac{1}{27\gamma}R^2\alpha\beta C_e^2$$

$$C = (1 - \gamma)RC_0 + \frac{2}{9\gamma}R\alpha C_e + \frac{1}{9\gamma}(3\gamma - 1)R\beta C_e$$

$$D = \frac{1}{27}(1 - \gamma)(3\gamma - 2)R^3\alpha\beta C_e^2(2C_0 + C_e)$$

$$E = \frac{2}{9}(1 - \gamma)(3\gamma - 2)R^2\alpha C_e(2C_0 + C_e) + \frac{1}{9}(1 - \gamma)(3\gamma - 1)R^2\beta C_e(2C_0 + C_e) + \frac{1}{27}R^2\alpha\beta C_e^2$$

$$F = \gamma(1 - \gamma)R(2C_0 + C_e) + \frac{2}{9}R\alpha C_e + \frac{2}{9}R\beta C_e.$$

VI. Acknowledgment

The authors wish to thank E. M. Billinghurst for his suggestions concerning some of the material in this paper, and to acknowledge the skill of R. C. McGee, without whose help this paper could not have been prepared.

Design of the ESIAC® Algebraic Computer*

M. L. MORGAN†, MEMBER, IRE, AND J. C. LOONEY‡, MEMBER, IRE

Summary—The concept of a pair of potential-plane "factor analogs," in which voltage measurements at the zeros and poles of a function are used for the calculation, is employed in the design of a general-purpose computer for algebraic functions of a complex variable. The logarithmic complex plane is used in order to represent a wide range of zeros and poles with uniform accuracy. Plotting facilities provide direct graphical output for applications such as frequency response plots and root-locus plots.

Introduction

THE CONCEPT of factor analog voltage distributions [1] makes possible for the first time a complete computer for algebraic functions of a complex variable based on pole-zero plots and potential analog pairs. Such functions occur in a great many practical engineering problems as a result of the widespread use of transforms for converting differential equations into algebric equations involving complex variables. Examples of such transform methods are equations involving the complex frequency $j\omega$, the Laplace transform, the Fourier transform, the D operator, or the z or w transforms used in analyzing sampled data systems. Through the use of such a transform, a differential equation is converted into an algebric equation involving a different variable and having a much easier solution. Once the solution is found, the inverse transform may be used to put the answer in terms of the original variable if desired. Often the characteristics of the transform function can be directly interpreted as the final practical answer, as in frequency response plots or root-locus plots.

The ESIAC®[1] Algebraic Computer is based on the principle of facter analog voltage distributions and designed for rapid calculation and plotting of rational algebraic functions and irrational and transcendental functions which can be expressed in the same general factor form.

The ESIAC Equations

A rational algebric function is often written as the ratio of two polynomials:

$$F(s) = \frac{a_0 + a_1 s + a_2 s^2 + \cdots + a_c s^c}{b_0 + b_1 s + b_2 s^2 + \cdots + b_d s^d}.$$

This equation can be factored to give either

$$F(s) = K_1 s^{n_0} \frac{(s - s_1)^{n_1}(s - s_3)^{n_3} \cdots}{(s - s_2)^{n_2}(s - s_4)^{n_4} \cdots},$$

where

$$K_1 = \frac{a_c}{b_d};$$

or

$$F(s) = K_2 s^{n_0} \frac{\left(1 - \frac{s}{s_1}\right)^{n_1}\left(1 - \frac{s}{s_3}\right)^{n_3} \cdots}{\left(1 - \frac{s}{s_2}\right)^{n_2}\left(1 - \frac{s}{s_4}\right)^{n_4} \cdots},$$

where

$$K_2 = \frac{a_0}{b_0}.$$

The above equations can be written in more compact form, and the logarithms of both sides can be taken and the real and imaginary parts separated for independent evaluation of the magnitude and the angle of the function. Using the symbol Π to represent a product of factors, $\sum$ to represent a summation of terms, $|\ |$ to denote magnitude and $\angle$ to denote angle, the equations are expressed as follows:

$$F = K_1 s^{n_0} \prod_i (s - s_i)^{n_i}, \tag{1}$$

or

$$\log |F| = \log |K_1| + n_0 \log |s| + \sum_i n_i \log |s - s_i| \tag{1a}$$

and

$$\angle F = \angle K_1 + n_0 \angle s + \sum_i n_i \angle (s - s_i). \tag{1b}$$

$$F = K_2 s^{n_0} \prod_i \left(1 - \frac{s}{s_i}\right)^{n_i}, \tag{2}$$

or

$$\log |F| = \log |K_2| + n_0 \log |s| + \sum_i n_i \log \left|1 - \frac{s}{s_i}\right| \tag{2a}$$

and

$$\angle F = \angle K_2 + n_0 \angle s + \sum_i n_i \angle \left(1 - \frac{s}{s_i}\right). \tag{2b}$$

In these equations F, s and the s_i's are complex quantities. For rational functions, K_1 or K_2 will be real and the n's will be integers, both positive and negative. The

* Received by the PGEC, August 20, 1960.
† Electro-Scientific Ind., Portland, Ore.
‡ Dept. of Elec. Engrg., Oregon State University, Corvallis, Ore.
[1] ®Trademark registered by Electro-Scientific Ind., Portland, Ore. U.S.A. and Foreign Patents Pending.

s_i values are the zeros and poles of the function—zeros in factors with positive exponents and poles in factors with negative exponents. The ESIAC handles functions expressed in the form of either (1) or (2), including functions which may have fractional exponents n.

Each complex quantity is represented in the ESIAC by its logarithmic components—that is, by a logarithmic magnitude scale and a linear angle scale. The s coordinate system is therefore the log s plane, rather than the s plane. The log s plane has a number of practical advantages over the s plane. Multiplying complex factors in log s coordinates might be likened to using a "two-dimensional slide rule." Like the slide rule, the log s plane has the useful characteristic that a given error in setting a position corresponds to a constant relative error (per cent or per unit error) anywhere in the coordinate system. Since the $|s|$ scale is logarithmic, its analogy with a slide rule is obvious; for the extension of the concept to the $\angle s$ scale, it should be noted that a small per cent error in $|s|$ corresponds to the same per cent of a radian. For example, an error of 1 per cent in $|s|$ represents the same displacement as an error in $\angle s$ of 0.01 radian.

Physical Description of the ESIAC

The ESIAC computer consists of a desk-sized console, shown in Fig. 1. There is a plotting table at the center and the various controls are grouped around it. On the right-hand panel is the basic equation, with controls for selecting the sign of K (corresponding to $\angle K = 0°$ or 180°), the exponent on $s(n_0)$ and the factor form $(s-s_i)^{n_i}$ or $(1-s/s_i)^{n_i}$. The center panel contains the null balance meters, the $|s|$ scale, the range switches for all the scales, and the mark selector switch. The range switches change numbers in windows on their scales, by powers of 10 on magnitude scales and multiples of 90° on angle scales. The holder for the graph paper is at the center of the table; to the left are the K and $\angle s$ scales and to the right the $|F|$ and $\angle F$ scales. The stylus for reading and marking the graph paper is mounted at the center of the handlebars used for positioning it. It is permanently coupled to the $|s|$ motion in the horizontal direction, but in the vertical direction it can be coupled to any of the other four scales: K, $\angle s$, $|F|$ or $\angle F$. For locating zeros and poles or for plotting root loci, the stylus will be coupled to $\angle s$; for frequency response plots it will be coupled to $|F|$ or $\angle F$.

The magnitude and angle analog sheets and their zero and pole contacts are located under covers at the left and the right, as shown in Fig. 2. Several probes are clamped to the moving frame with their tips sliding on the sheets. The supply of probes and probe clamps is stored in the drawer at the lower left. Fig. 3 shows the method of placing a probe tip contact to represent a zero or pole. The moving frame can be thought of as a coordinate system relative to which the center of the sheet corresponds to the readings on the $|s|$ and $\angle s$ scales. The frame is constrained to move in translation only, in the manner of a drafting machine, and the $|s|$ and $\angle s$ scales are permanently coupled to its motion. To find the location for a probe tip to represent a particular zero or pole s_i, all that is necessary is to set $s = s_i$, place the tip of the probe in the indented index point at the center of the sheet, and clamp the probe to the frame, as shown. The probe clamping screw contains an electrical admittance element (a capacitor) representing the exponent n_i; probe clamps are available for various values. The threaded sockets for the probe clamps occur in pairs, the upper hole connecting to the positive bus bar (for zeros) and the lower to the negative bus bar (for poles). Both the upper and lower frame members contain a pair of bus bars, and 50 pairs of sockets are provided for zeros and poles over each sheet. The probe bodies are slotted for versatility in

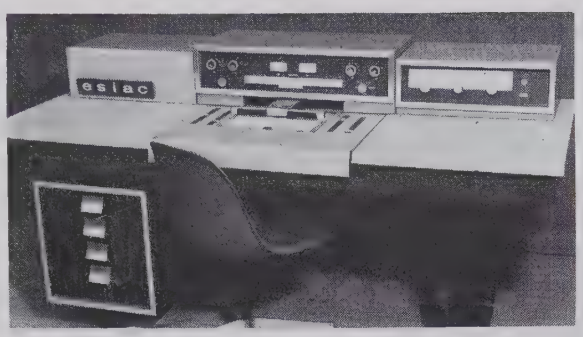


Fig. 1—The ESIAC.

Fig. 2—Operating controls and analog sheets.

Fig. 3—Placing a probe.

placement. When there are too many zeros and poles in a small area, extra tips can be mounted with insulating washers on probes already placed, and connected electrically by extension leads to probe clamps in any available bus bar sockets.

Using the ESIAC

The procedure for solving any equation on the ESIAC is to set first the given or assumed quantities in the equation, then to adjust the controls for the unknown to obtain a null balance and read or plot the setting. The automatic mark circuit can be set to mark the graph paper at the stylus whenever a solution of the magnitude or angle equation is passed, thus automatically drawing a curve such as the root-locus plot in Fig. 4. To find fixed roots, the locus for the given $\angle F$ will be drawn, as shown; then it will be retraced to find the points which show a magnitude balance. In feedback control system design it is frequently desired to mark the whole locus with values of magnitude (gain) which will place the root at various points along the curve, as shown in Fig. 5. The automatic mark circuit can also be used to draw the magnitude loci intersecting the angle loci. If it is desired, curves can be drawn for various settings of $|F|$ and $\angle F$ to produce a whole set of orthogonal curves to picture the function as a field.

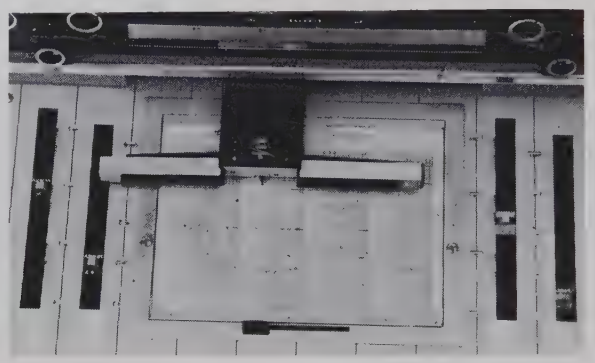


Fig. 4—Curve plotting.

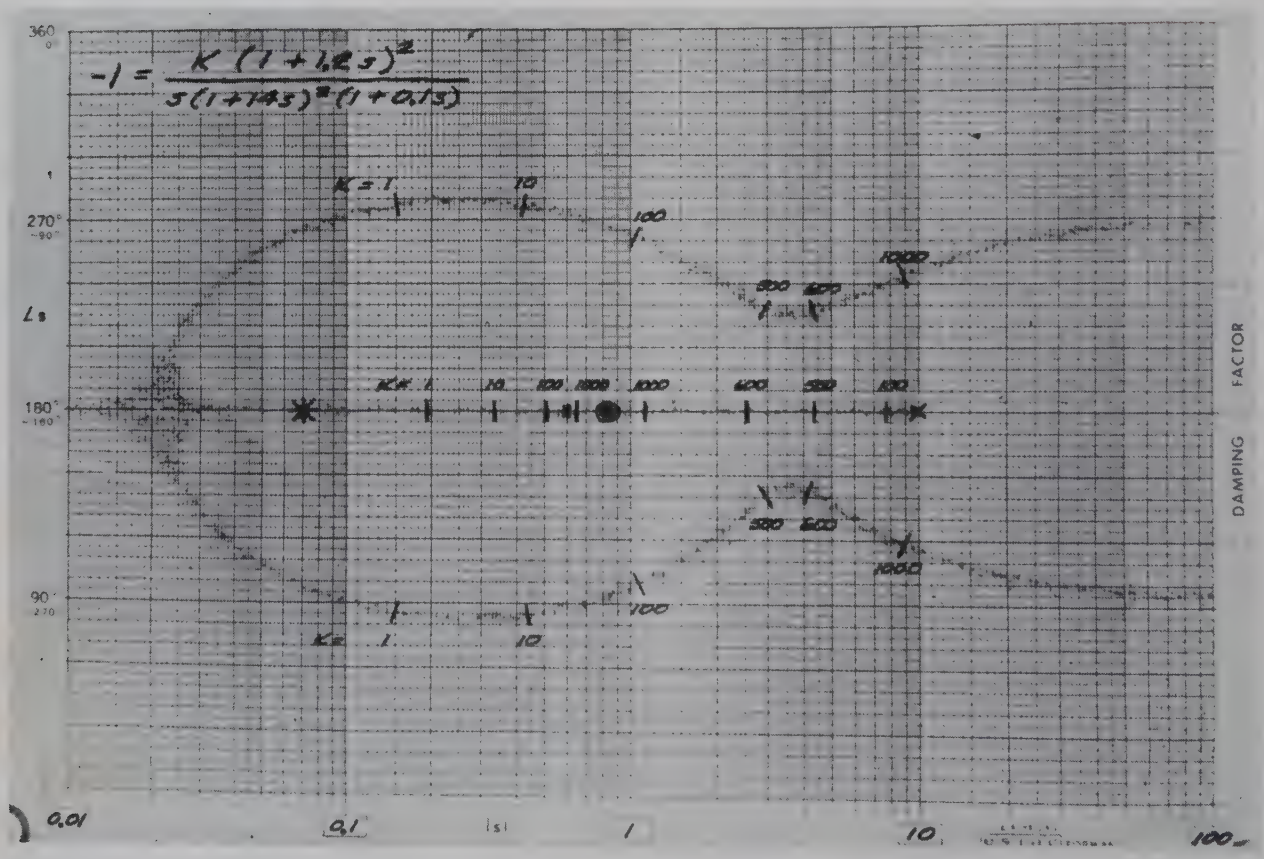


Fig. 5—An ESIAC root-locus plot.

Calculating Circuits

Fig. 6 is a basic block diagram of the calculating circuits of the ESIAC. Two separate circuits are used: one for the magnitude equation (1a) or (2a), and the other for the angle equation (1b) or (2b). The quantities log $|s-s_i|$ or log $|1-s/s_i|$ are voltages measured simultaneously at all the s_i points (the zeros and poles) by probe contacts touching a voltage distribution called the "magnitude sheet." The quantities $\angle(s-s_i)$ or $\angle(1-s/s_i)$ are similarly measured on another distribution called the "angle sheet." The choice between factor forms is provided by a change in power supply connections to the sheets.

Each s_i voltage from a sheet is applied to an admittance (the capacitor in the probe clamp) proportional to the corresponding n_i. The currents representing positive terms (zeros) are collected in a bus bar connected to one input of a differential current amplifier, and those representing negative terms (poles) in a bus bar connected to the other input. Currents representing the log magnitudes and the angles of F, K and s^{n_0} are set by calibrated controls operating precision voltage dividers. These currents are permanently connected to the proper bus bars. The $|s|$ and $\angle s$ controls also set the coordinates of the sheet position in addition to operating their respective voltage dividers. A null balance on the magnitude meter indicates a solution of the magnitude equation; an angle balance, a solution of the angle equation. When both meters show a null simultaneously, a solution of the complex equation has been found.

Since $\angle F$ readings differing by 360° represent the same complex quantity, the $\angle F$ dial covers only a 360° range. Net current representing more than 360° may be encountered, however; therefore a bidirectional stepping relay automatically inserts multiples of 360° as required to keep the unbalance less than 360°, thus ensuring that all solutions of the equation will be found without attention from the operator.

Although the theory of factor analog distributions can be described most easily in terms of dc voltage, ac is

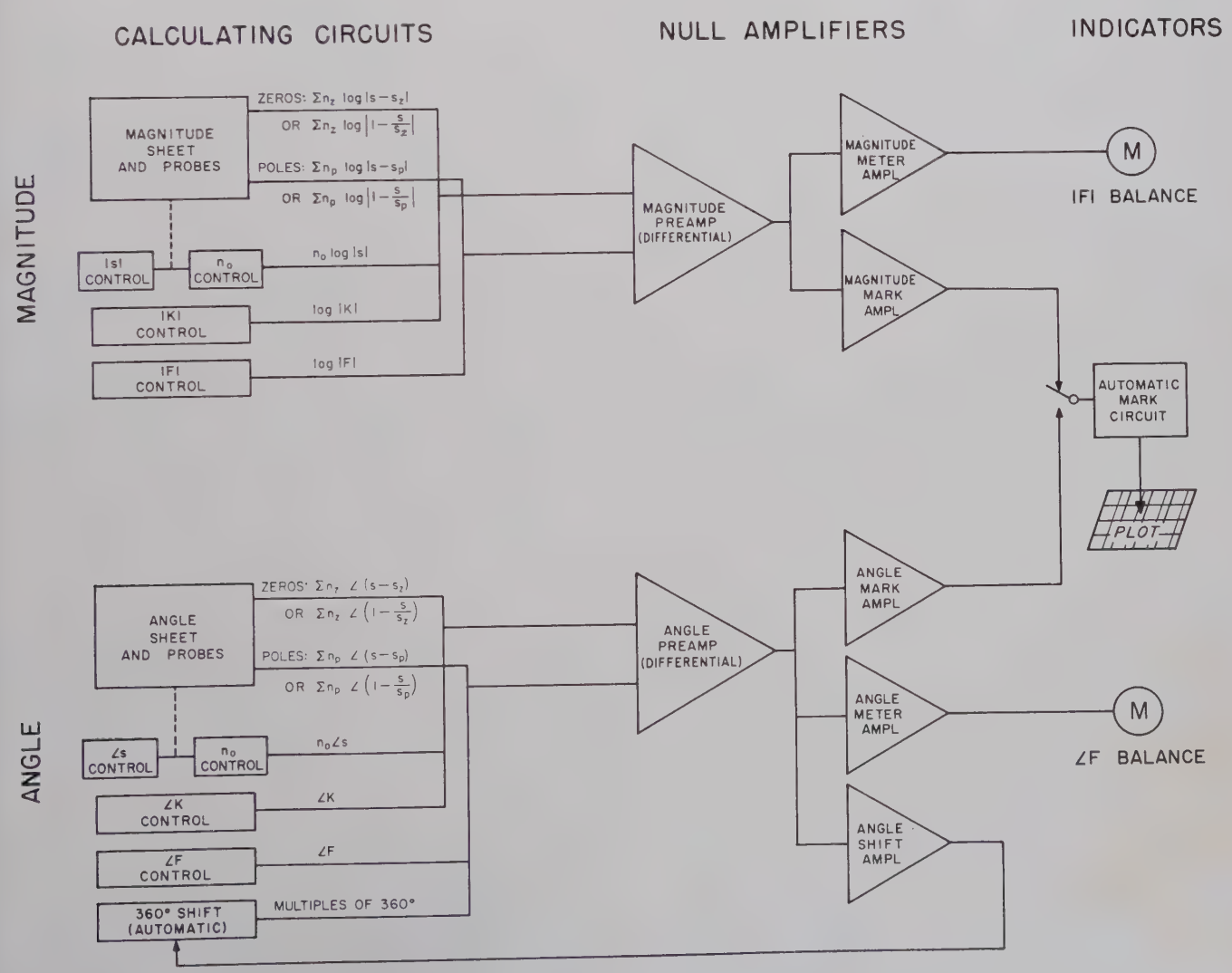


Fig. 6—Block diagram of calculating circuits.

actually used. There are a number of practical advantages for ac, including the simplicity of a transformer power supply without rectifiers and filters, the maintenance of accurate voltage ratios, the isolation of windings, the availability of accurate taps for range switching, the better stability and easier adjustment of small capacitors compared to large resistors for representing exponents, and the well-known advantages of sensitive ac null detectors over high-gain chopper-stabilized dc amplifiers.

Fig. 7 shows the sheet power supply connections for the two alternate factor forms. In this figure the approximate fields are sketched with solid lines for streamlines and dotted lines for equipotentials. It will be noticed that ac voltage is used rather than dc, but that plus and minus signs are retained since polarity is significant in the calculation. Another difference between this figure and the basic factor analog sketches in the article on the basic theory of factor analog computing [1] is that the fields have been reversed, that is, rotated 180°. The voltage distributions in this article are shown as they are viewed by the operator of the ESIAC. The reason for the reversal is that the ESIAC sheets are mounted stationary and a frame carrying the probes moves as s changes. When $|s|$ and $\angle s$ approach their maximum values, in the upper right-hand corner of the graph paper, the frame also moves in the same direction. The center of the sheet is then at the lower left-hand corner of the frame, so it can be seen that the zero and pole contact configuration in the frame and the voltage distribution on the sheet appear upside down relative to the configuration on the graph paper.

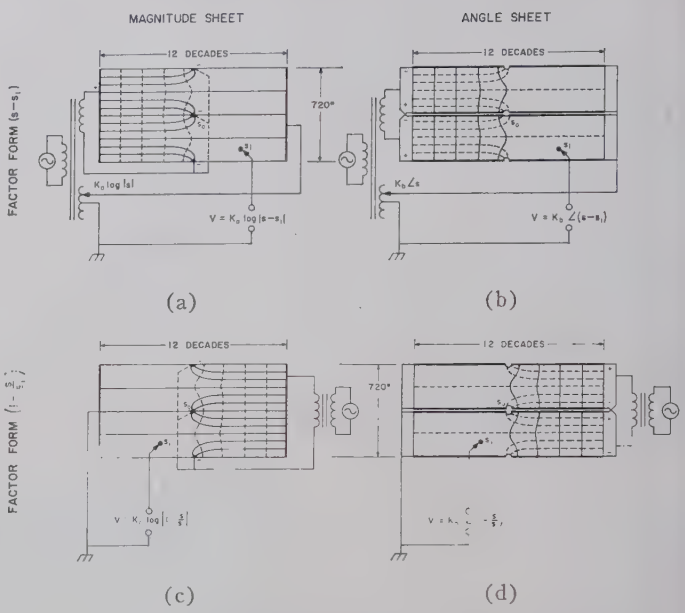


Fig. 7—Analog sheet connections.

Fig. 8 is a simplified schematic diagram of the calculating circuits for the factor form $(1-s/s_i)$. The switch which changes the sheet connections to the other factor form is omitted for simplification. There are many other simplifications in the diagram, such as the omission of the range switching associated with the various potentiometers.

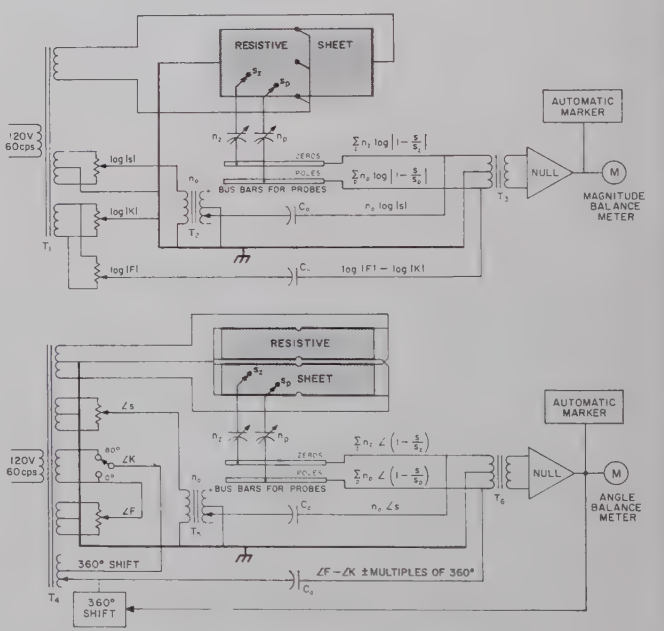


Fig. 8—Simplified circuit schematic for factor form $(1-s/s_i)$.

Null Amplifiers

Two nearly identical, transistorized null amplifiers are used in the ESIAC; one for the magnitude circuit and one for the angle circuit. The only difference is the additional angle output amplifier to operate the 360° shift circuit. Each null amplifier consists of a preamplifier followed by parallel amplifier channels, as shown in Fig. 6. The output of one channel for each circuit is connected to a synchronous demodulator circuit to provide polarity-sensitive meter null indications for visual null balancing. The output of the second channel is connected to the automatic marking circuit for use in plotting solutions.

The input signal to the null amplifiers has a very large dynamic range. It can vary rapidly from 1 μv to 1 v. It was necessary to design this amplifier so that it would overload without the meter doubling back and without spurious automatic marking as a result of distortion at very high inputs. The amplifier must also recover from heavy overloads practically instantaneously so that spurious nulls will not be plotted during the recovery period. This was accomplished by a symmetrical nonlinear feedback-type clipping circuit around each stage [3].

During rapid scanning of the log s plane, the automatic mark circuit must respond to nulls almost instantaneously; therefore, a rectifier and filter cannot be used to inhibit the mark. Instead, it is necessary to sample the signal at the peak of each cycle and then mark the paper if the level is below the desired threshold. The circuit of Fig. 9 performs this operation. The gating signal holds the thyratron cut off all the time ex-

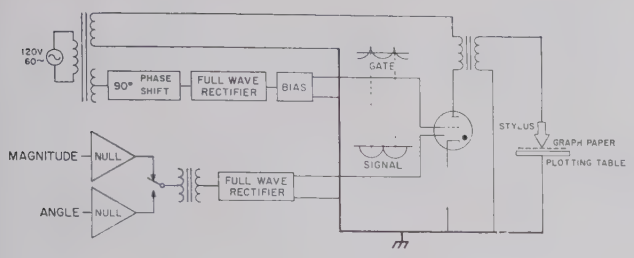


Fig. 9—Automatic mark circuit.

cept for about 1 msec at the peak of the signal. If the signal is absent at this instant, the thyratron fires and generates a pulse in the output transformer which marks a spot on the graph paper.

Accuracy

The transformer ratios and capacitor values in the ESIAC calculating circuits are accurate to better than 0.1 per cent, and the effective accuracy of each potentiometer and dial is on the order of 0.01 inch on the scale. The principal accuracy limitation at present is the difficulty of obtaining sheet material of uniform resistivity, combined with the requirements for long wear, low noise, etc.

Sheet accuracy can be expressed as the maximum displacement of voltage contours from their theoretical positions. This displacement is equivalent to the displacement of a zero or pole plotted in the log s plane. The Teledeltos®[2] paper sheets used in the first ESIAC computers have been accurate to about 0.10 inch on the magnitude sheet and 0.05 inch on the angle sheet. Research in progress on better resistive sheet materials is promising. The sheets have been designed as plug-in units so that as better materials become available the old sheets can be replaced to achieve the accuracy for which the rest of the ESIAC was designed.

The effect of equivalent zero and pole displacement error on the accuracy of the answer must be determined for each function at the particular value of the solution, since it is a function of the partial derivatives of the function with respect to the zeros and poles. About the only generalization worthwhile is the observation that when s is larger than the majority of the zeros and poles, the factor form $(s-s_i)$ is more accurate, since s predominates over the s_i's in these factors. When s is smaller than the majority of the zeros and poles, the factor form $(1-s/s_i)$ is better, since these factors approach unity.

[2] ® Trademark registered by Western Union Telegraph Co.

Applications

The first ESIAC computers produced have already proved to be of great value in speeding up the solution of a variety of problems in amplifier and network calculation, and especially in feedback control system design [2]. Among the most popular applications are:

1) Root-locus plotting
2) Frequency response plotting
3) Polynomial factoring
4) Residue evaluation

The versatility of the ESIAC is further enhanced by the fact that it can be used with irrational functions and with transformations such as the z plane and the w plane used in the analysis of sampled data systems. The exponential functions encountered in problems with dead time or transport lag are easily approximated, and nonlinear problems can frequently be handled by the describing-function method. In general, the ESIAC is useful for nearly all problems in which differential equations in time can be represented either exactly or approximately by any of the algebraic transform methods.

References

[1] M. L. Morgan, "Algebraic function calculations using potential analog pairs," Proc. IRE, vol. 49, pp. 276–282; January, 1961.
[2] M. L. Morgan, "A new computer for algebraic functions of a complex variable," *Proc. First IFAC Moscow Congress*, Moscow, Russia, to be published by Butterworths Scientific Publications, London, England; 1961.
[3] J. C. Looney, "Designing amplifiers with nonlinear feedback," *Electronics*, vol. 34, pp. 46–49; March 31, 1961.

Correspondence

A Basic Limitation on the Speed of Digital Computers*

The design of digital computers is fast approaching the state in which computer speed is restricted by the speed of light. For example, a computer with a one-nsec (one billionth of a second) access time must have an average radius of less than one foot simply because, at the speed of light, it takes at least one nsec for a piece of information to be transmitted over a distance of one foot.

With this restriction in mind, one can reach the conclusion that computers will have to be made smaller and smaller in order to continue the trend toward more speed However, this trend cannot continue indefinitely; there is an inherent quantum-mechanical limit to the speed obtainable. In fact, it is shown below that the access time t_a must be greater than

$$\sqrt[4]{N/d}\ 10^{-20}\ \text{sec},$$

where N is the number of bits of randomly accessed storage, and d is the average density (in grams/cm^3) of the material used in constructing the memory.

Following the lead of Bremermann[1] we invoke the Heisenberg uncertainty principle to assert that a physical measurement must be limited by the relation[2]

$$\Delta E \cdot \Delta t \geq h/2\pi, \tag{1}$$

where h is Planck's constant, ΔE is the uncertainty in energy and Δt is the uncertainty in time.[3] Thus for a "bit" of information which requires e ergs of energy and is to be accessed (measured) in t_a seconds, we have

$$e \cdot t_a \geq h/2\pi. \tag{2}$$

Suppose we have an "optimized" computer which consists of a central processing unit surrounded by a spherical memory storage unit with radius r, with average density d, and with maximum access time t_a.

If M is the total mass of the memory storage unit then

$$M \leq (4/3)\pi r^3 d.$$

The energy equivalent is

$$E = Mc^2 \leq (4/3)\pi r^3 d c^2,$$

where c is the speed of light ($\sim 3\times 10^{10}$ cm/sec). The energy e per bit is then

$$e = E/N \leq \frac{4}{3}\frac{\pi r^3 d c^2}{N}, \tag{3}$$

which represents an upper bound on the amount of energy which can be utilized for storing each bit of information.

In order for the computer to operate correctly, it cannot make errors in measurement of energy which exceed e, and cannot make errors in time which exceed the access time t_a. Thus, by (2),

$$h \leq 2\pi e \cdot t_a \leq \frac{8}{3}\frac{\pi^2 r^3 d c^2}{N} \cdot t_a. \tag{4}$$

We seek now a relation between r and t_a. Since a signal must travel a distance r for each (maximum) access, we have

$$r \leq c \cdot t_a, \tag{5}$$

where c is the velocity of light. Thus (4) becomes

$$h \leq \frac{8}{3}\frac{\pi^2 t_a^3 c^3 d c^2}{N} \cdot t_a = \frac{8}{3}\frac{\pi^2 c^5 d t_a^4}{N},$$

or

$$N \leq K d t_a^4, \tag{6}$$

and

$$t_a \geq \sqrt[4]{N/d}\ K^{-1/4} \tag{7}$$

where

$$K = \frac{8}{3}\frac{\pi^2 c^5}{h} \simeq 3 \cdot 10^{79} < 10^{80}.$$

Thus

$$t_a \geq \sqrt[4]{N/d}\ 10^{-20}. \tag{8}$$

Example: If $d = 20$ grams/cm and $N = 10^9$ then

$$t_a \geq \sqrt[4]{\frac{10^9}{3 \cdot 10^{79} \cdot 20}}\ \text{sec} > 10^{-18}\ \text{sec}.$$

Thus a computer with a billion bits of storage can have an access time of no less than 10^{-18} seconds, unless materials are used whose density is greater than 20 grams/cm^3. The access time t_a is bounded below by 10^{-21} sec, even when N takes the minimum value of 2 (assuming $d \leq 60$).

An obvious way in which the access time can be lowered, without sacrificing storage capacity, is to increase the density of the materials in the memory storage unit. However, because of the 4th root in the right-hand side of (8), a large increase in d will result in a much smaller decrease in t_a.

The restriction (5), which is inherent in present-day serial computers where all information is passed through a central processing unit, might be somewhat overcome in truly "parallel" machines.

Thus, a digital computer of the future might contain a great many "local" processing units *distributed uniformly* throughout the memory storage area and under the general control of a central processing unit.

W. W. Bledsoe
Advanced Research
Palo Alto, Calif.

* Received by the PGEC, March 18, 1961; revised manuscript received, April 3, 1961.

[1] H. J. Bremermann, "A quantum-theoretic limitation on physical representation of information," submitted to Proc. IRE for publication.

[2] L. I. Shiff, "Quantum mechanics," McGraw-Hill Book Co., Inc., New York, N. Y., p. 7; 1955.

[3] Eq. (1) is only an approximation to the uncertainty principle which is given more precisely in terms of expected values of certain probability distributions.

Axiomatic Majority-Decision Logic*

In a recent paper,[1] Cohn and Lindaman omitted a step in their derivation of (12), the K equation of the Majority-Logic Binary Full Adder. The omission of the step does not effect an incorrect result, but does mislead the reader, thus casting doubt on the ease of application of the basic axioms and theorems. It is not true that Line 3 in the derivation can be obtained by applying Theorem 13 to Line 2. Theorem 8 must first be applied to Line 2, and then Theorem 13 may be used. The derivation with the missing step follows. All equation numbers are as in the original paper.

$K = [(A \# 0 \# B) \# 1 \# (A \# 0 \# C)]$
$\quad \# 1 \# (B \# 0 \# C).$
Application of T9 (with $W = A$, $X = B$, $Y = C$, $Z = 0$) yields
$K = [A \# 0 \# (B \# 1 \# C)] \# 1 \# (B \# 0 \# C).$
Application of T8b with $W = 0$, $X = A$, $Y = (B \# 1 \# C)$, $Z = (B \# 0 \# C)$ yields
$K = [(B \# 0 \# C) \# (B \# 1 \# C) \# A]$
$\quad \# 1 \# (B \# 0 \# C).$
Application of T13 (with $W = B$, $X = C$, $Y = 0$, $Z = 1$, $Q = A$) to the expression in braces yields
$K = [B \# C \# (1 \# 0 \# A)] \# 1 \# (B \# 0 \# C).$
Finally, as in the original paper,
$K = (A \# B \# C) \# 1 \# (B \# 0 \# C)$ (by T4)
$\quad = B \# C \# (A \# 0 \# 1)$ (by T13)
$\quad = A \# B \# C$ (by T4).

John P. Larkin
I.T.T. Federal Labs.
Nutley, N. J.

* Received by the PGEC, March 22, 1961.

[1] M. Cohn and R. Lindaman, "Axiomatic majority-decision logic," IRE Trans. on Electronic Computers, vol. EC-10, pp. 17–21; March, 1961.

A Modulo Two Adder for Three Inputs Using a Single Tunnel Diode*

The following designs exploit only the essential aspects of the tunnel diode's voltage-current characteristic; any negative resistance device could therefore be substituted for the diode. A piece-wise linear approximation of the characteristic is shown in Fig. 1 (not to scale) with three load lines and four stable operating points. Fig. 2 gives the circuit of a modulo two adder, operating in accordance with these states and load lines.

* Received by the PGEC, May 22, 1961.

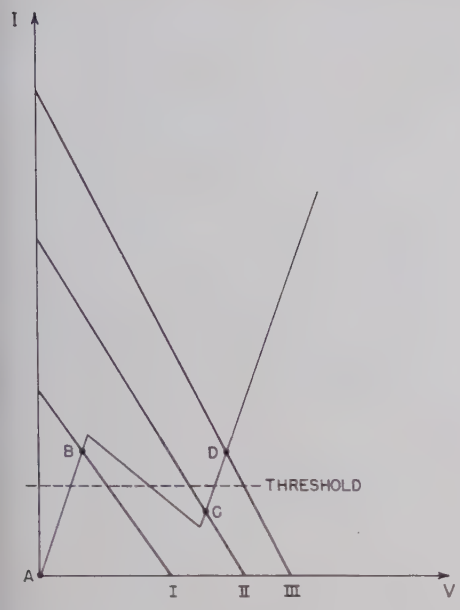


Fig. 1.

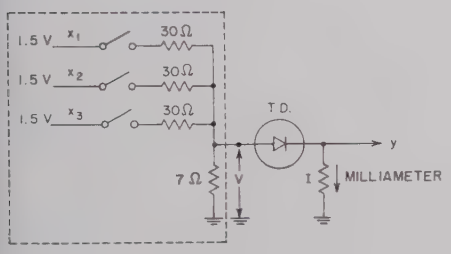


Fig. 2.

TABLE I

V volts dc	x_1 x_2 x_3 0 = low (open) 1 = high (closed)	y 0 = low ($I \leq 4.7$ ma) 1 = high ($I > 4.7$ ma)	I ma dc
0	0 0 0	0	0
0.25	0 0 1	1	6.2
0.25	0 1 0	1	6.2
0.46	0 1 1	0	3.3
0.25	1 0 0	1	6.2
0.46	1 0 1	0	3.3
0.46	1 1 0	0	3.3
0.60	1 1 1	1	5.9

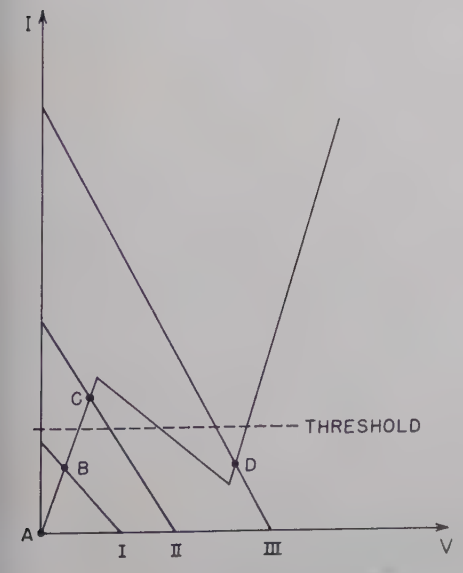


Fig 3.

Resistances of the circuit are such that:

1) with all switches open, the diode is at state A and the output current is low (zero);
2) with exactly one of the three switches closed, the Thévenin equivalent of the linear input circuit (shown dotted in Fig. 2) operates on load line I, hence the diode rests at state B, and the output current is low;
3) with exactly two of the switches closed, operation is on load line II at state C, and the output is high;
4) with all three switches closed, operation is on load line III at state D, and the output is high.

The measured outputs for the 2^3 possible switch settings of Fig. 2 are listed in Table I. From this table, it is apparent that output y is high if and only if an odd number of inputs are high, *i.e.*, $y=(x_1+x_2+x_3)$ mod 2.

A wide variety of linear input circuits may be used provided that the three relevant load lines intersect the diode characteristic at states B, C, D, respectively, and at no other point. Thus, with this single limitation, arbitrary (and not necessarily identical) current or voltage sources may be substituted for those of Fig. 2.

Theoretically, the "threshold current" above which the output is "high" and below which the output is "low" may be chosen anywhere between the state C value and the minimum of B and D. Practical considerations suggest that the threshold be close to the mean of these two values. Ideally, the state B value equals the state D value, but there is nothing in the design which assures that such is the case. In fact, the states are rather independently sensitive to variations in the input circuit parameters as well as to fluctuation of the diode characteristic due to environmental variations. Circuit failure results when the state C current exceeds the threshold or when the B or D current does not.

Other logic is obtainable from such negative resistance devices employed in the same circuit configuration. Obviously, the elimination of one input (say) x_3 results in two-input mod 2 addition, *i.e.*, $y=(x_1+x_2)$ mod 2 which is also call EXCLUSIVE OR logic. Biasing one of the latter two inputs (say) x_2 gives an inverter, *i.e.*, $y=(x_1+1)$ mod $2=\bar{x}_1$. Furthermore, relocation of the load lines can give essentially different symmetric logic; *e.g.*, the four states selected in Fig. 3 are such that the corresponding circuit has a high output current if and only if exactly two of the three inputs are high. Note that all states must be on portions of the diode characteristic having positive slope, in order that the inherent instability of the negative resistance region be avoided.

Finally, these design principles may be extended to symmetric logic accommodating more than three input lines by placing tunnel diodes in series. The composite characteristic of n diodes in series has n distinct negative resistance regions; hence n diodes in series form the basis for a mod 2 adder accommodating $2n+1$ inputs.

KARL S. MENGER
Harvard Computation Lab.
Cambridge, Mass.

Minimizing Incompletely Specified Sequential Switching Functions*

Paull and Unger[1] have given a very neat method for finding the minimal equivalents of finite-state machines whose flow tables are incompletely specified with respect to next-states or outputs or both. The results they obtain are probably as complete as one can hope for without going in for actual enumeration. However, their analysis of "incompletely specified functions"[2] is somewhat ambiguous and leaves one with the impression that incomplete specification of next-states and that of the outputs are of equal intrinsic significance to the minimization problem. It is the purpose of this note to comment on this and point out some interesting connections between the results of Paull and Unger[1] and some published by us elsewhere on the synthesis of finite-state machines. It is hoped that this brief note will complement the basic contribution of Paull and Unger to the synthesis problem. (For the sake of clarity, we shall use the notations and terminology of Paull and Unger freely.)

It is easily verified that to every flow table containing unspecified next-states, one can construct an equivalent flow table where all the unspecified next-states are marked F (for "forbidden state") and an additional row—corresponding to the state F—is added to the flow table. In each column, for F, the next-state is F and the output is unspecified (*i.e.*, $N(F, Ik)=F$ for each k and $Z(F, Ik)$ is unspecified for each k). Now, clearly, F is output compatible with each internal state and from the way $N(F, Ik)$ is specified and from Theorem 3 of Paull and Unger,[1] it follows that F is compatible with each internal state. In other words, F is contained in every member of any closed C-set. It is seen thus that unspecified next-states do not have intrinsic significance to the determination of C-sets with the closure property (which is the core of the reduction problem). Any reduction procedure valid for flow tables with only unspecified outputs will be equally valid for a flow table with both next-states and outputs unspecified or even only next-states unspecified. Whether an actual reduction of rows is possible or not will of course depend on the way the notion of "covering" is defined. In the augmented flow table, F, by definition, being a member of every compatible, it is unnecessary to carry it along during the reduction procedure and one may proceed to form the compatibles ignoring F altogether. All this is, of course, implicit in the treatment of the problem in Paull and Unger[1] but our main concern here is to clarify the relative roles played by the next-states and outputs in incompletely specified functions.

The way we have defined F, above, and the augmented flow table suggests itself naturally if we consider the problem of specification of a sequential machine in terms of input sequences (events) and outputs. (A passing reference is made to this in Paull

* Received by the PGEC, February 4, 1960.

[1] M. C. Paull and S. H. Unger, "Minimizing the number of states in incompletely specified sequential switching functions," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-8, pp. 356–367; September, 1959.

[2] *Ibid.*, paragraph 2.

and Unger.[2]) Unspecified next-states (in the flow table) always arise when restrictions are placed on the allowed input sequences. In all such events one may assume that the machine enters a forbidden state (F) in which it remains indefinitely. It is sufficient to have a single forbidden state to take into account all unallowed input sequences. From this it would seem that the proper point of departure for the synthesis of a finite-state machine is not the flow table but its specification in terms of events and outputs. It is unfortunate that, out of purely historical reasons, this intrinsic connection between specification and reduction has not been studied with greater perception.

An attempt to consider the synthesis problem in its completeness (in the sense indicated above) was made by us some time ago.[3] There, starting from the specification of a machine in terms of events and outputs, a normal form of its transition table (the flow table of Paull and Unger[1]) is constructed and a method is outlined for its reduction to a minimal machine. The reduction procedure is in two parts. In the first, the transition table is modified by assigning superscripts (Greek letters are used as superscript symbols) to the internal states, all output compatible states being assigned the same superscript. Next, an iterative algorithm is given to modify these superscripts systematically so that the original output compatible sets are partitioned into sets which are now also implication compatible. It will be seen that the procedure outlined is quite similar to that given in Appendix II of Paull and Unger.[1]

Three distinct cases are considered: 1) where the event-output specification is complete; 2) where some events are not assigned outputs and 3) where restrictions/equivalences of input sequences are prescribed with or without "don't-care" outputs. The transition tables associated with case 3) are the same as the (augmented) incomplete flow tables discussed earlier in this note.

The treatment in Srinivasan and Narasimhan[3] is of course "classical" (in the terminology of Paull and Unger[1]) and only non-overlapping partitions are considered. Hence, the solution suggested to case 2) is the trivial one of enumeration. However, it is perhaps of some interest to note that that reduction algorithm, with only minor modifications, can be used to generate overlapping compatibles. We should like, here, to outline very briefly the modifications, so that this modified process can be used as an alternative to the processes systematized in Paull and Unger.[1] (We shall use the terminology of Srinivasan and Narasimhan[3], and assume familiarity with the procedure outlined there.) The reduction is still carried out in 2 stages. Only, now, the modified transition table is formed by assigning *multiple* superscripts (*i.e.*, strings of Greek letters; we shall refer to these as the *indices*) to the states according to the following scheme: States ($Si_1, \cdots, Si_m$) have a common symbol in their indices if they are output compatible. This assignment of indices

	I_1	I_2	I_3	I_4
1	$\alpha\beta$	1, 2, 3, 8, 9		
2	$\alpha\beta$	1, 2, 6		
3	$\alpha\beta$		1, 3, 4	
4	α	4, 7		
5	$\alpha\beta$	3, 4, 5, 8, 9		
6	β	5, 6		
7	$\alpha\beta$		2, 7, 6, 8, 9	
8	α			4, 8
9	β			6, 9

(a)

	I_1	I_2	I_3	I_4
1	$\alpha\alpha_1\beta\beta_1$	1, 2, 3, 8, 9		
2	$\alpha\alpha_1\beta\beta_1$	1, 2, 6		
3	$\alpha\alpha_1\beta\beta_1$		1, 3, 4	
4	α	4, 7		
5	$\alpha_1\beta$	3, 4, 5, 8, 9		
6	β	5, 6		
7	$\alpha\beta_1$		2, 7, 6, 8, 9	
8	$\alpha\alpha_1$			4, 8
9	$\beta\beta_1$			6, 9

(b)

Fig. 1—The reduction table. (a) At the start. (b) At the end. Refers to flow table XIV(a) of Paull and Unger.[1] The compatibles are: α(123478); α_1(12358); β(123569); β_1(12379).

can be systematized by a method completely analogous to that explained in Srinivasan and Narasimhan.[4]

The reduction table is now formed exactly as in Srinivasan and Narasimhan;[3] only, the row-names and the states in the cells now have indices associated with them rather than single superscripts. (Note that it is not necessary to include F in this table either as a row name or as a cell member.) Call two indices incompatible if they have *no* symbol in common. The reduction is now carried through, referring to the row indices, by repeated application of the following rule: Find 2 cells, in one column, with states having a common symbol, say α, in their indices but whose row indices are incompatible. Then change all the α's in one of the cells to α_{i+1} (where α up to α_i already occur as index symbols). Add the new symbol α_{i+1} to the indices of all states, *not included in these 2 cells*, containing the symbol α. Modify the row indices to conform to these changes and repeat the procedure. Clearly this iteration can be carried through in terms of the minor and major cycles as indicated in Srinivasan and Narasimhan.[3] The process is complete when no new symbol is added in one complete major cycle. The compatibles now correspond to the distinct symbols in the indices, all states having that symbol in their index forming one compatible. Using these compatibles, the minimization can be completed as in Paull and Unger.[1]

As an example, Fig. 1(a) and 1(b) gives the reduction table at the start and end of the reduction, for the flow table XIV(a).[1] (In order not to clutter up the tables, the indices inside the cells have not been indicated.)

In concluding this note, perhaps it may be pointed out that not sufficient attention seems to have been paid to the close analogy that exists in the minimization techniques of combinational and sequential switching functions. A technique for the former in terms of "covering" tables has been worked out by Svoboda and a modification of his procedure to take into account "don't-care" states has been suggested by Nadler.[5] This latter is very similar to the overlapping subdivision scheme systematized in Paull and Unger.[1]

R. Narasimhan
29/A, Edward Elliot Rd.
Mylapore, Madras, India

[5] M. Nadler, "Some topics on the design of switching circuits" (mimeographed notes), Tata Inst. of Fundamental Res., Bombay, India; 1959.

High-Speed Transistorized Adders*

The undersigned noticed with interest an article[1] on a "High-Speed Transistorized Adder for a Digital Computer," by Forest Salter.

We have written papers on a very similar circuit, and the following are references to them:

1) "Parallel addition in digital computers—a new fast carry circuit," *Proc. IEE*, vol. 106, pt. B, pp. 464–466; September, 1959.

2) Discussion on paper entitled "Elimination of carry propagation in digital computers," *Proc. Internatl. Conf. on Information Processing*, Paris, France, p. 394; June 15–20, 1959.

3) "A parallel arithmetic unit using a saturated transistor fast carry circuit," *Proc. IEE*, vol. 107, pt. B, pp. 573–584; November, 1960.

D. B. G. Edwards
Elec. Engrg. Labs.
The University
Manchester, England

[3] C. V. Srinivasan and R. Narasimhan, "On the synthesis of finite sequential machines," *Proc. Indian Acad. Sci.*, vol. 50, pp. 68–82; July, 1959.

[4] *Ibid.*, paragraph 5.

* Received by the PGEC, June 26, 1961.

[1] IRE Trans. on Electronic Computers, vol. EC-9, pp. 461–464; December, 1960.

Analog Computation of Covariance Matrices*

An analog computation technique is developed which permits the rapid evaluation of the elements in the covariance matrix of the time-dependent (transient) output of a linear system which is excited by Markoffian noise. This note discusses the basic approach and derives the underlying mathematical relationship. The method has been applied in practice and has been found to be an effective, flexible, and accurate means of obtaining covariance matrices.

To carry out the above program we first recall a few facts concerning time-dependent variances. Briefly, let $x(t)$ belong to a stationary stochastic process with mean zero and correlation function $\psi(t-t') = Ex(t)x(t')$. Then if $x(t)$ is passed through a linear system with impulsive response $h(t, \xi)$, the output noise is given by

$$y(t) = \int_0^t h(t, \xi)x(\xi)d\xi$$

and its variance at time t is

$$Ey^2(t) = \int_0^t \int_0^t h(t, \xi)h(t, \zeta)\psi(\xi - \zeta)d\xi d\zeta. \quad (1)$$

It is desirable to compute the time-dependent variance $Ey^2(t)$ on an analog computer. If $\psi(\xi-\zeta)$ is of exponential form (a commonly encountered type of correlation function) and the system is time independent, then (1) may be readily simulated.[1] The crucial result that allows an easy analog mechanization is the mathematical formula

$$\int_0^t \int_0^t h(t-\xi)h(t-\zeta)e^{-\beta|\xi-\zeta|}d\xi d\zeta = F^2(t) + 2\beta\int_0^t F^2(u)du \quad (2)$$

where

$$F(z) = \int_0^z h(z-x)e^{-\beta x}dx. \quad (3)$$

Now in more sophisticated statistical problems it is necessary not only to compute the variance $Ey^2(t)$ of the output, but the *covariances* $Ey(t)y(t')$ as well. Analogous to (2), the crucial mathematical formula (derived in the Appendix) is

$$\int_0^t d\xi \int_0^{t'} d\zeta h(t-\xi)h(t'-\zeta)e^{-\beta|\xi-\zeta|} = F(t)F(t') + 2\beta\int_0^t F(u)F(u+t'-t)du, \quad t' \geq t \quad (4)$$

where $F(z)$ is defined by (3).

The operations indicated by (4) may be represented in block diagram form as shown in Fig. 1. It is a simple matter to translate the scheme of the block diagram into an analog computer circuit as soon as the transfer function $H(p)$ is specified.

The computer's output is obtained as a continuous plot of $N(t, t')$ vs t' (and t) for one

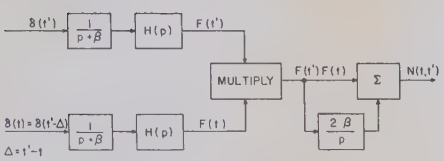


Fig. 1—Block diagram of method for computing covariance values in the transient case.

specific value of $\Delta = t'-t$. Numerical values of $N(t', t)$ may be selected from this curve for specified t and t'. If we let t and t' take on the discrete values $t_k = kT$, $1 \leqq k \leqq m$ then the elements $N(t_1, t_{\alpha+1})$, $N(t_2, t_{\alpha+2})$, $\cdots$, $N(t_{m-\alpha}, t_m)$ of the covariance matrix $N = \|N(t_j, t_k)\|_{1\leq j,\ k\leq m}$ may be selected from the curve for which $t'-t = \alpha T$. The values from any single curve lie on a diagonal parallel to the main diagonal of the matrix. (The terms on the main diagonal are variances and are readily obtained from this simulation by running the case of zero delay; that is $t = t'$.) Because the matrix N is symmetric, $N(t, t') = N(t', t)$, it is only necessary to compute $N(t, t')$ for $t' \geqq t$.

Appendix

We wish to prove (4). Let

$$I = \int_0^t d\xi \int_0^t d\zeta h(t-\xi)h(t'-\zeta)e^{-\beta|\xi-\zeta|}.$$

Without loss of generality we may assume $t' \geqq t$. If we make the change of variable $t-\xi = u$, $t'-\zeta = v$ (which has unit Jacobian) in the above integral, then I becomes

$$I = \int_0^t du \int_0^{t'} dv h(u)h(v)e^{-\beta|u+\Delta-v|}$$

where $\Delta = t'-t \geqq 0$. The above integral may be written as

$$\begin{aligned} I &= \int\int_{v>u+\Delta} h(u)h(v)e^{-\beta(v-u-\Delta)}dudv \\ &\quad + \int\int_{v<u+\Delta}^{h(u)h(v)} e^{-\beta(u+\Delta-v)}dudv \\ &= \int_\Delta^{t+\Delta} h(v)dv \int_0^{v-\Delta} h(u)e^{-\beta(v-u-\Delta)}du \\ &\quad + \int_0^t h(u)du \int_0^{u+\Delta} h(v)e^{-\beta(u+\Delta-v)}dv. \end{aligned}$$

The advantage of this manipulation is that it eliminates the unpleasant absolute-value sign in the exponent of $e^{-\beta|u+\Delta-v|}$. Now make the change of variable $v-\Delta = y$ in the first integral and the change of notation $u = y$ in the second. Then

$$I = \int_0^t [h(y+\Delta)F(y) + h(y)F(y+\Delta)]dy \quad (5)$$

where we have introduced

$$\begin{aligned} F(z) &= \int_0^z h(z-x)e^{-\beta x}dx \\ &= \int_0^z h(w)e^{-\beta(z-w)}dw. \end{aligned}$$

We may also write $F(z)$ as

$$F(z) = e^{-\beta z}g(z)$$

where

$$g(z) = \int^z h(w)e^{\beta w}dw. \quad (6)$$

Writing F in terms of g in (5) leads to

$$I = \int_0^t [h(y+\Delta)g(y) + h(y)g(y+\Delta)e^{-\beta\Delta}]e^{-\beta y}dy. \quad (7)$$

But from (6), $g'(z) = h(z)e^{\beta z}$. Thus, we may write (7) as

$$\begin{aligned} I &= e^{-\beta\Delta}\int_0^t [g'(y+\Delta)g(y) + g'(y)g(y+\Delta)]e^{-2\beta y}dy \\ &= e^{-\beta\Delta}\int_0^t e^{-2\beta y}\frac{d}{dy}[g(y+\Delta)g(y)]dy. \quad (8) \end{aligned}$$

Integrating by parts and using the fact that $g(0) = 0$ immediately reduces (8) to (4).

T. W. Connolly
Sperry Gyroscope Co.
Syosset, N. Y.
K. S. Miller
Dept. of Math.
New York University
New York, N. Y.

Multichannel Minimum-Amplitude Comparator*

Introduction

Often, in dealing with both digital and analog circuitry, where many simultaneous data-channel signals are present, one encounters the problem of finding (at some preselected time) the data channel which contains the minimum-amplitude signal.

In an analog device this minimum-amplitude channel may represent the line with the maximum correlation between an incoming signal and a correlation matrix. It might also represent the channel with the minimum deviation from a given set of quiescent conditions. In fact, there are many devices where a minimum is often more desirable than a maximum.

In quasi analog-to-digital equipment where pulse amplitude represents the amount of deviation from a standard (such as in character-recognition devices), minimum-pulse amplitude detection is not only desirable, but also is absolutely necessary. For example, in the field of character recognition, a scanned character may match a standard character in 94 of a possible 96 points, whereas the same scanned character may match another standard character in 93 of the same 96 points. A typical example of this is the similarity between the letters *C* and *O*. One can see the difficulty in "recognizing" the character with the best possible match. There is only a 1 per cent difference between 93 and 94. However, if one looks for the character with the minimum mismatch it is then only a problem of detecting between 2 and 3 mismatches, or a difference of 33 per cent. It was for problems such as these that the multichannel minimum-

* Received by the PGEC, December 10, 1960; revised manuscript received, April 10, 1961.

[1] J. H. Laning and R. H. Battin, "Random Processes in Automatic Control," McGraw-Hill Book Co., Inc., New York, N. Y., ch. 6; 1956.

* Received by the PGEC, May 12, 1961.

amplitude comparator was devised. This letter discusses the digital or minimum-pulse application. The same type of discussion holds true for the analog application.

Circuit Operation

The circuit for the comparator is shown in Fig. 1. Each channel has a comparator transistor Q_i associated with it. The voltage output e_i of each channel is applied to the base of its transistor Q_i through its source impedance R_s. The transistors are normally cut off (in the nonconducting state). The emitters of the comparator transistors are brought together at point E and are fed by a positive "interrogation" or "selector" pulse through a series capacitor-resistor combination R-C. The channel voltages e_i are pulses, occurring at time t_o, and may vary from zero to plus 20 v, the range being limited by the maximum voltage ratings of the transistor being used. These pulses are in the direction to keep the transistor in the normally *cutoff* state. At some time t_d+t_o, the interrogation pulse is applied in the direction to turn *on* all the comparator transistors. The transistor with the minimum voltage e_j applied to its base will turn on first, clamping point E to the voltage e_j. All the other $n-1$ transistors will remain "off," as their base-to-emitter voltages will be in the direction to keep them in the nonconducting state.

The current I is then steered into the comparator transistor associated with the correct channel, that is, the one whose voltage output is a minimum. This current either can be used to provide a voltage output across R_L, or can be steered through a magnetic-core code converter, which converts the n-bit channel code into a more compact code.

Circuit Considerations

I_{eo}-Temperature Problem

Point E, the common emitter point, initially is at a fixed voltage $-V_e$ such that the n transistors are cut off. The total reverse current I_{eo} flows into this point.

If this point were to be initially clamped by a diode to the cutoff voltage $-V_e$, then upon application of the interrogation pulse the diode would be released and the interrogation pulse would have to supply the total reverse current. Assume that a current I_{ej} is required to supply the load requirements of the "recognized" transistor. Under a large temperature rise, I_{eo} might increase to the point where

$$I - I_{eo} < I_{ej}. \tag{1}$$

Under these conditions the circuit will be inoperative because the load requirements of Q_j will not be satisfied. By the use of the inductor L, the steady-state reverse current will flow in the inductor, and if L is large enough, will continue to flow in the inductor during the time of the interrogation pulse. The interrogation pulse would then no longer have to supply the temperature-dependent reverse current.

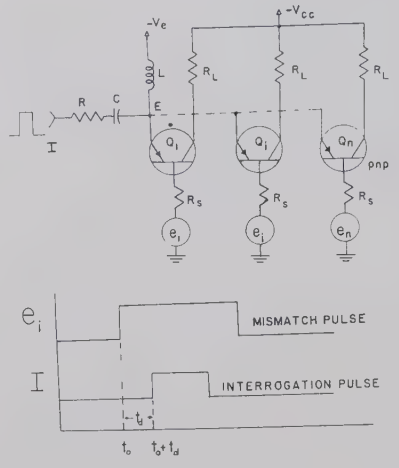


Fig. 1—Multichannel minimum-amplitude comparator.

Mode of Operation

The question of in what mode this circuit should be operated brings out some interesting answers. At first glance, the circuit designer would like to operate this circuit in the switching mode. That is, all the "unrecognized" transistors are cut off and the "recognized" transistor is in saturation or "full on."

Examine, however, this mode of operation. For the transistor Q_j to switch, the drive current I must be adjusted for the worst condition, that is e_j at its maximum.

$$I > \frac{V_{cc} + e_j}{R_L}. \tag{2}$$

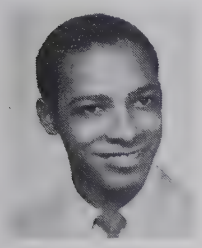
However, when e_j is at its minimum, the excess drive current, above that required by the load, flows into the base circuit of the "recognized" transistor. This excess base current causes a voltage drop across R_s, effectively raising the minimum voltage e_j. It is quite possible that the minimum voltage e_j may be raised to the point where it is no longer the minimum voltage and misreading occurs. *Therefore, switching of the recognized transistor is definitely not desirable.* The load resistors R_L should be kept small, so that the operating point of the "recognized" transistor is kept in the constant-current portion of the transistor characteristics, *i.e.*, an almost vertical load line.

This mode of operation is ideal for driving inductive loads, such as a magnetic-core code converter, as the transistor is a constant current source. Due to the fact that the "recognized" transistor is operating in the grounded base configuration, this circuit is capable of operating at high frequencies. This circuit has been operated at 1 Mc with ease, and tests indicate that it can be pushed even higher. The multichannel comparator has been used in up to 100 different channels and will sense a differential of as small as 50 mv. This comparator is a neat, compact, and extremely reliable solution to the problem of detecting minimum voltage among a given set of varying voltages. Additional circuit sensitivity can be built into such a device by using the output collector swing of the recognized transistor to decrease its input base voltage.

Rob Roy
Dept. of Elec. Engrg.
Rensselaer Polytechnic Institute
Troy, N. Y.

Contributors

G. A. Alphonse (S'56–M'59) was born in Port-au-Prince, Haiti, on December 22, 1935. He came to the United States in September, 1954, to study at New York University, New York, where he received the B.E.E. degree in 1958, and the M.E.E. degree in 1959.



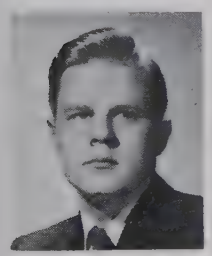
G. A. Alphonse

While at N.Y.U., he did fundamental research in solid-state devices with emphasis on superconductive devices. In 1959 he joined the staff of the RCA Laboratories, Princeton, N. J., where he has continued his work on superconductive devices.

Mr. Alphonse is a member of Tau Beta Pi and Eta Kappa Nu.

❖

Algirdas Avizienis (S'55–M'56) was born on July 8, 1932, in Kaunas, Lithuania. He received the B.S., the M.S., and the Ph.D. degrees in electrical engineering from the University of Illinois, Urbana, in 1954, 1955, and 1960, respectively.



A. Avizienis

In 1955, he was employed as a research engineer at the Jet Propulsion Laboratory, California Institute of Technology, Pasadena. In 1956, he resumed graduate studies at the Digital Computer Laboratory of the University of Illinois. Here he held University Fellowships in 1954–1955 and 1956–1957, and the RCA Fellowship in 1957–1958; he also was Research Assistant at the Digital Computer Laboratory in 1959–1960 and staff engineer at Barnes and Reinecke, Inc., Chicago, Ill., in 1958–1960. Upon completion of graduate studies, he returned to the Jet Propulsion Laboratory, where he is senior research engineer in the Guidance Computers Section.

Dr. Avizienis is a member of Sigma Xi, Tau Beta Pi, Eta Kappa Nu, Pi Mu Epsilon, and the ACM.

❖

William A. Barrett was born in Central City, Neb., on November 8, 1930. He received the B.S. and M.S. degrees in physics and mathematics from the University of Nebraska, Lincoln, in 1952 and 1953, respectively, and the Ph.D. degree in physics from the University of Utah, Salt Lake City, in 1957. His Ph.D. dissertation work was on the capture of micromesons in iron.

Since 1957, he has been a member of the technical staff in the Device Development Department of Bell Telephone Laboratories, Murray Hill, N. J., where his work has included the development of twistor devices.

W. A. Barrett

Dr. Barrett is a member of the American Physical Society, Sigma Xi, and Phi Beta Kappa.

❖

William G. Brown was born in Chicago, Ill., on May 1, 1929. He received the B.S. degree in engineering mathematics from The University of Michigan, Ann Arbor, in 1951.

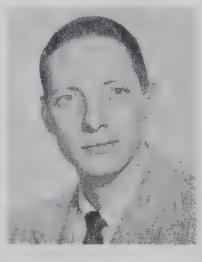
W. G. Brown

From 1951 to 1955 he was a graduate research assistant at the Willow Run Laboratories, The University of Michigan, where he was responsible for the logic design, debugging, and testing of a high-speed digital computer for multiple-output real-time control applications, and where he also performed the system design of an integrated surveillance data processor for a field army, and designed a general-purpose educational computer for The University of Michigan. He joined Hermes Electronics Company, Cambridge, Mass. (formerly Hycon Eastern), Inc.), in 1955 as a Senior Logical Designer, where he performed the system study and logic design of a telegraph automatic switching system employing magnetic-drum message storage, and where he also performed the mathematical analysis of practical redundant-circuit models for improving digital system reliability, and designed time-generator and tape-search units for use in data reduction systems. Since 1957, he has been with Cook Electric Company, Morton Grove, Ill., where he is providing project engineering and project consulting services to a number of programs requiring the analysis and design of digital processing systems.

Mr. Brown is a member of Tau Beta Pi and Phi Kappa Phi.

❖

John A. Brussolo (SM'58) was born in Bisbee, Ariz., on March 31, 1930. He attended the California Institute of Technology, Pasadena, from 1948 to 1950, and served in the U. S. Navy from 1951 to 1955. He received the B.S.E.E. degree in 1958 and the M.S. degree in 1959, both from the University of California at Berkeley.



J. A. Brussolo

From 1958–1960, he was a teaching and research assistant at the University of California. Currently he is employed by the Berkeley Division of Beckman Instruments, Inc., Richmond, Calif., where he is engaged in the design and development of analog computer systems.

Mr. Brussolo is a Registered Engineer in the state of California and a member of AIEE, Eta Kappa Nu, Tau Beta Pi, Sigma Xi, and Phi Beta Kappa.

❖

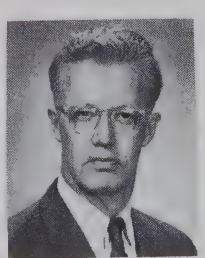
L. L. Burns, Jr. (S'43–A'44–M'48–SM'52) was born in Houston, Tex., on June 12, 1923. He received the B.S. degree in 1943 from the A & M College of Texas, College Station, and the M.S. degree in 1952 from Princeton University, Princeton, N. J. After serving as an officer with the Signal Corps for three years, he joined the RCA Laboratories, Princeton, N. J., in 1946. His initial work there was on solid-state phenomena and integrated devices. Later, he worked on color television, and for the past three years has been with the Computer and Digital Systems Laboratory as manager of the cryoelectric program.

L. L. Burns, Jr.

Mr. Burns is a member of AIEE, and Sigma Xi.

❖

Robert P. Coleman (M'59) was born in Salt Lake City, Utah, on June 7, 1909. He received the B.S. degree in science from the California Institute of Technology, Pasadena, in 1931, and the Ph.D. degree in physics from Yale University New Haven, Conn., in 1946.



R. P. Coleman

From 1935 to 1947, he was employed as a physicist by the National Advisory Committee for Aeronautics (now NASA) at the Langley Laboratory, Hampton, Va., where he conducted experimental and theoretical research in aircraft flutter, vibration, and aerodynamics.

His work on the theory of self-excited mechanical oscillations of hinged rotor blades explained the mechanism of dangerous "ground resonance" in helicopters and predicted the conditions for its occurrence.

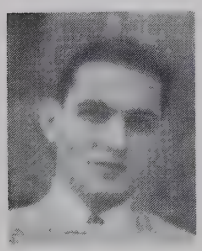
From 1947 to 1955, Dr. Coleman was employed by the Franklin Institute Laboratories for Research and Development, Philadelphia, Pa., as Project Leader, Section Chief, and Senior Staff Physicist. His work was concerned with applications of theoretical physics and mathematics to the detection of land mines, gust alleviation for aircraft, basic radar studies, thermoelectricity, and conduction and noise in resistors.

In September, 1955, he joined Burroughs Corporation, Paoli, Pa., as Head of the Circuits Group in the Tubes, Circuits, and Components Department, and his work was concerned with the theory of ferroresonant circuits. In 1956, he was transferred to the Applied Physics Department to work on the theory of switching of magnetic cores. He has since served as consultant in the Systems Engineering Department and the Systems Research Department, working primarily on theoretical problems in character and pattern recognition and in switching theory.

Dr. Coleman is a member of Sigma Xi, the Scientific Research Society of America, and the AAAS.

❖

Benjamin M. Eisenstadt (A'52–M'57) was born in Philadelphia, Pa., on April 9, 1925. He received the B.S.E.E. and M.S.E.E. degrees from the Massachusetts Institute of Technology, Cambridge, in 1951.



B. M. Eisenstadt

From 1942 to 1944, he was employed by the U. S. Army Signal Corps, and during the following two years served in the U. S. Navy, being discharged as a radio technician. In 1951, he became a staff member of the M.I.T. Lincoln Laboratory, Lexington, in the Long Range Communications Group. From 1956 to 1959, he was employed by the Hermes Electronics Company, Cambridge, where he was a member of the Communications Group. Since 1959, he has been employed by the Applied Research Laboratory, Sylvania Electric Products, Inc., Waltham, Mass., working in the Information Processing Research Department.

❖

Herbert L. Gelernter was born in Brooklyn, N. Y., on December 17, 1929. He received the B.S. degree, *summa cum laude*, in physics from Brooklyn College, N. Y., in 1951, and the Ph.D. degree in theoretical physics from the University of Rochester, Rochester, N. Y., in 1956.

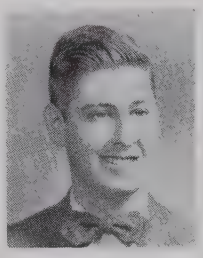
Since 1956, he has been a member of the research staff of the International Business Machine Corporation, Yorktown Heights, N. Y. During this time, he has been concerned with problems of machine intelligence, pattern recognition, non-numerical data processing, and advanced data-reduction techniques, for experimental physics. He is currently on leave of absence, serving as visiting fellow at the European Organization for Nuclear Research (CERN) in Geneva, Switzerland.

H. Gelernter

Dr. Gelernter is a member of Phi Beta Kappa, Sigma Xi, Pi Mu Epsilon, The American Physical Society, and the ACM.

❖

Arthur Gill was born in Haifa, Israel, on April 18, 1930. He received the B.S.E.E. and M.S.E.E. degrees from the Massachusetts Institute of Technology, Cambridge, in 1955 and 1956, respectively, and the Ph.D. degree in electrical engineering from the University of California, Berkeley, in 1959.



A. Gill

From 1954 to 1956, he was a Teaching Assistant in the Department of Electrical Engineering at M.I.T. From 1956 to 1957, he worked in the Research Division of the Raytheon Manufacturing Company, Waltham, Mass., where he was engaged primarily with semiconductor circuitry design. Since 1957, he has been at the University of California, Berkeley, as a Teaching Associate in electrical engineering and more recently as an Assistant Professor. He is also associated with the Electronic Research Laboratory, where he is working on information theory problems, and with the Advanced Development Group of the Bendix Computer Division of Bendix Aviation Corporation.

Dr. Gill is a member of Eta Kappa Nu, Tau Beta Pi, and Sigma Xi.

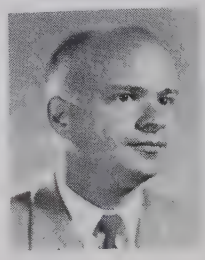
❖

Bernard Gold (SM'59) was born in New York, N. Y., on March 31, 1923. He received the D.E.E. degree from the Polytechnic Institute of Brooklyn, Brooklyn, N. Y., in 1948.

B. Gold

From 1950 to 1953, he was with the Hughes Aircraft Company, Culver City, Calif. Since 1953 he has been with the Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, working in the fields of radar, communications, and pattern recognition.

Wilbur H. Highleyman (S'55–M'58) was born in Kansas City, Mo., on August 20, 1933. He received the B.E.E. degree in 1955 from Rensselaer Polytechnic Institute, Troy, N. Y., the M.S. degree in 1957 from The Massachusetts Institute of Technology, Cambridge, and the D.E.E. degree in 1961 from the Polytechnic Institute of Brooklyn, N. Y.



W. H. Highleyman

From 1955 to 1956, he was a research assistant at M.I.T. Lincoln Laboratory, Lexington, developing transistor circuits for digital computers. The following year, he continued work in transistor circuits and thin ferromagnetic films at M.I.T. on a fellowship from Melpar, Inc.

In 1958 he joined the Bell Telephone Laboratories, Murray Hill, N. J., where he was engaged in the problem of character recognition until 1959. Since then, he has been concerned with the development of data communication equipment and the study of new devices for data communication problems.

Mr. Highleyman is a member of Tau Beta Pi and Eta Kappa Nu, and an associate member of Sigma Xi.

❖

Floyd B. Humphrey (M'58) was born in Greeley, Colo., on May 20, 1925. He received the B.S. and Ph.D. degrees in physical chemistry from the California Institute of Technology, Pasadena, in 1950 and 1956, respectively.

F. B. Humphrey

In 1955, he joined the Solid-State Device Development Department of the Bell Telephone Laboratories, Murray Hill, N. J., where he studied the relation between the structure and the magnetic characteristics of ferrite. Later, he investigated the mechanism of magnetization-flux reversal in thin ferro-magnetic films. Finally, he supervised the development of a permanent magnet memory. In 1960, he joined the Guidance and Control Research Section, Jet Propulsion Laboratory, California Institute of Technology, where he is concerned with the solid-state physics of computer devices.

Dr. Humphrey is a member of American Physical Society and the American Rocket Society.

❖

Louis A. Kamentsky (S'51–M'57) was born in Newark, N. J., on July 28, 1930. He received the B.S.E.E. degree from the Newark College of Engineering in 1952 and the Ph.D. degree in engineering physics

from Cornell University, Ithaca, N. Y., in 1956.

He held a research assistantship at Cornell from 1952–1954. From 1954–1955, he was on the staff of the Columbia University Electronics Research Laboratory, New York, N. Y., where he made studies of the fundamental accuracy of radar systems. From 1955–1956, he held an RCA Fellowship at Cornell. In 1956, he joined Bell Telephone Laboratories, Murray Hill, N. J., where he worked in a digital computer systems group; one of his functions there was to develop character recognitions systems. Since 1960, he has been with IBM Research Center, Yorktown Heights, N. Y., workong on image processing systems.

L. A. Kamentsky

Dr. Kamentsky is a member of Sigma Xi, Tau Beta Pi, Eta Kappa Nu, and Omicron Delta Kappa.

❖

William H. Kautz (S'46–A'52–M'57) was born in Seattle, Wash., on February 9, 1924. He received the B.S.E.E. and M.S.E.E. degrees in 1948, the M.S. degree in mathematics in 1949, and the D.Sc degree in electrical engineering in 1951, all from the Massachusetts Institute of Technology, Cambridge.

W. H. Kautz

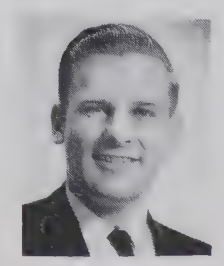
From 1949 to 1951, he was a Research Assistant in the M.I.T. Research Laboratory of Electronics. In 1951, he joined the staff of the Stanford Research Institute, Menlo Park, Calif., where he is currently a Senior Research Engineer in the Computer Techniques Laboratory. He has been engaged in the logical design and testing of digital systems, the development of techniques for the analysis and synthesis of switching networks, and the development of codes and coding systems for computers and communications systems.

Dr. Kautz is a member of the American Mathematical Society, Sigma Xi, the Association for Symbolic Logic, the Society for Industrial and Applied Mathematics, and the ACM.

❖

George Kovatch (S'59) was born in Sranton, Pa., on February 20, 1934. He received the B.S.E. degree from Princeton University, Princeton, N. J., in 1955, and the M.S. degree from Cornell University, Ithaca, N. Y., in 1960. Currently, he is a Ph.D. candidate at Cornell University.

From June, 1955, until February, 1956, he was a member of the General Electric Company's Engineering Training Program. He served as a First Lieutenant in the Air Force from February, 1956, until November, 1957, as a Communications Officer. In December, 1957, he rejoined GE as a participant in the company's Honor's Program until February, 1960.



G. Kovatch

He worked as an electrical engineer in Advanced Circuits and Devices Engineering at the GE Light Military Electronics Department's Advanced Electronics Center, Cornell University Industry Research Park, Ithaca, until September, 1960, in electronics, research and development, which included work on magnetic and semiconductor circuits and devices, and the development of a solid-state commutator motor for space vehicle applications. He is continuing his studies at Cornell with a fellowship from the Foundation for Instrumentation, Education, and Research, sponsored by Electronics Associates, Inc.

Mr. Kovatch is a member of Sigma Xi and the AIEE

❖

G. W. Leck was born in Indianapolis, Ind., on September 8, 1908. He received the E.E. degree from the Drexel Institute of Technology, Philaphia, Pa., in 1936.

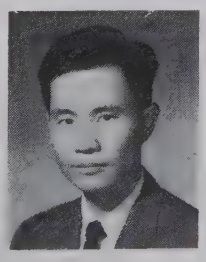
G. W. Leck

He then joined the research group of the RCA Manufacturing Company, Camden, N. J., where his initial work was on active microwave devices. In 1942, he transferred to the Research Department of the RCA Laboratories, Princeton, N. J., where he was active on devices involving energy level transitions and atomic phenomena. More recently, he has also done work on active thin-film solid-state devices for computer applications. Presently, he is doing research on thin-film cryoelectric devices.

Mr. Leck is a member of Sigma Xi.

❖

C. Y. Lee (S'47–A'50–M'55) was born in Shanghai, China, on December 10, 1926. He received the B.E.E. degree from Cornell University, Ithaca, N.Y., in 1947, and the M.S.E.E. degree and Ph.D. degree in mathematics from the University of Washington, Seattle, in 1949 and 1954, respectively. His main interests up to then were in functions of a complex variable.



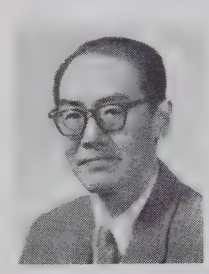
C. Y. Lee

From 1952 to the present, he has been a member of the technical staff with the Bell Telephone Laboratories, Inc., Whippany, N. J. From 1952 to 1957, he worked on switching theory and the design of switching networks. He spent the academic year 1957 to 1958 as a visiting member of the Institute for Advanced Study, Princeton, N. J., in the School of Mathematics. While there, he worked on the theory of Turing machines and self-organizing systems. Since 1958, he has been working in the field of automata and programming theory and the design of computers for formal analysis and theorem-proving processes.

Dr. Lee is a member of the American Mathematical Society and the ACM.

❖

Arthur W. Lo (S'43–A'50–SM'56) was born in Shanghai, China, on May 21, 1916. He received the B.S. degree in physics from Yenching University, China, in 1938, the M.S. degree in physics from Oberlin College, Oberlin, Ohio, in 1946, and the Ph.D. degree in electrical engineering from the University of Illinois, Urbana, in 1949.



A. W. Lo

He taught physics and electronics in several colleges, both in this country and abroad, before he joined the Radio Corporation of America, Princeton, N. J., in 1951. There, he developed a number of basic techniques of employing solid-state devices (magnetics, semiconductors and optoelectrics) for logic and memory, including the transfluxor. He was in charge of a switching research group responsible for developing various basic millimicrosecond switching techniques, using parametric phase-locked oscillators and tunnel diodes. In 1960, he joined the International Business Machines Corporation, Poughkeepsie, N. Y., where he is presently conducting work on exploratory digital techniques.

Dr. Lo is a member of Sigma Xi.

❖

James C. Looney (S'53–A'54–M'56) was born in McMinnville, Ore., on July 29, 1929. He received the B.S.E.E. and M.S.E.E. degrees from Oregon State University, Corvallis, in 1954 and 1960, respectively.

J. C. Looney

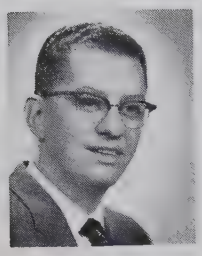
From 1954 to 1956, he was a member of the technical staff of Bell Telephone Laboratories, Whippany, N. J., where he participated in the graduate study program and worked on transistor switching circuits and analog-digital conversion techniques. He was employed by Tektronix, Inc., Portland, Ore., from 1956 to 1957. In 1957, he joined the faculty of Ore-

gon State University, where he is presently an Assistant Professor in electrical engineering. He also serves as a consultant to Electro Scientific Industries, Inc., Portland.

Mr. Looney is a member of Tau Beta Pi, Phi Kappa Phi, and the Professional Engineers of Oregon.

❖

Herman Lukoff (A'52–M'57–SM'58) was born in Philadelphia, Pa., on May 2, 1923. He received the B.S.E.E. degree in 1943 from the Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia.

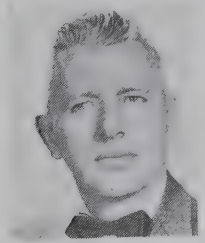


H. Lukoff

After graduation, he was involved in circuit development on the ENIAC project. From 1944 to 1946, he engaged in radio-radar maintenance in the U. S. Navy, after which he joined the EDVAC project at the University of Pennsylvania. He joined the Electronic Control Company, later incorporated as the Eckert-Mauchly Computer Corporation, in 1947, and was a member of the Univac staff engaged in the design of BINAC and UNIVAC circuits. When Eckert-Mauchly joined Remington Rand, Philadelphia, he was successively promoted to UNIVAC I Project Engineer in charge of a group of engineers and technicians responsible for the test and delivery of the first 10 UNIVAC I systems to customers. In 1955, he was named LARC Project Coordinator, and in 1956, an Engineering Director. He has been with the LARC Project from its beginning stages to the present, and is now Manager, UNIVAC Engineering Center.

❖

Donald H. MacPherson (SM'57) was born in Summit, N. J., on November 23, 1919. He received the B.E.E. degree from Manhattan College, Bronx, N. Y., in 1942.



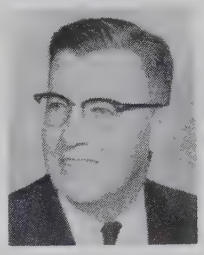
D. H. MacPherson

In 1942, he joined the Naval Ordnance Laboratory, where he was engaged in the development of Naval Magnetic Mines. From 1943 to 1946, he served as a commissioned Naval Officer engaged in the development of special anti-submarine weapons. In 1946, he joined Bell Telephone Laboratories, Murray Hill, N. J., as a member of the System Development Department. After completing a training program, he was engaged in logic and circuit design on a number of projects, including nationwide toll dialing, automatic circuits for subscriber line insulation resistance measurements and a concentrator for telephone answering service. In addition, he supervised a group on the logic and circuit design of the Sage System common user group air-ground communication network. In 1956, he transferred to the Special System Development Department where he has since been engaged in supervising a group on exploratory and specific circuit development employing magnetic logic and digital storage.

Mr. MacPherson is a member of Eta Kappa Nu.

❖

Wilbur E. Meserve (SM'57) was born on January 4, 1901, in Gorham, Me. He received the B.S.E.E. degree and M.S. degree in physics, from the University of Maine, Orono, in 1923 and 1926, respectively, and the M.E.E. and Ph.D. degrees from Cornell University, Ithaca, N. Y., in 1929 and 1933, respectively.



W. E. Meserve

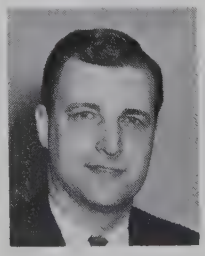
He then joined the Electrical Engineering Department of Cornell, where he served as a member of the faculty, primarily in the area of feedback control systems, for thirty years. During this period, he also served as consultant to the General Electric Company, Stromberg-Carlson, Cornell Aeronautical Laboratory, Project Lincoln at M.I.T., and other universities.

During 1955, he served as special lecturer and consultant in automatic control systems under a Fulbright grant at the University of Sydney and the New South Wales University of Technology, Sydney, Australia.

Dr. Meserve is a member of Sigma Xi, Tau Beta Pi, ASEE, and AIEE.

❖

Merle L. Morgan (S'48–A'54–M'60) was born in Whittier, Calif., on May 28, 1919. He received the B.S. degree, the M.S. degree, and the Ph.D. degree in electrical engineering in 1949, 1950, and 1954, respectively, from the California Institute of Technology, Pasadena. He conceived the idea for the ESIAC while a graduate student and wrote his Ph.D. dissertation on this topic.



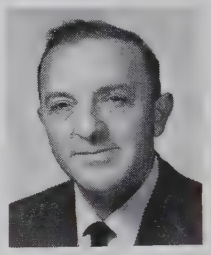
M. L. Morgan

During World War II, he served in Civilian Public Service under the direction of the U. S. Forest Service, where he collected and analyzed experimental data and designed instrumentation. From 1946 to 1948, he was with the McCullough Tool Company in Los Angeles, Calif., designing electronic circuits for use in oil well logging and servicing. At the completion of his graduate studies in 1954, he joined the staff of Electro Scientific Industries, Inc., Portland, Ore., where he is Director of Research and Engineering.

Dr. Morgan is a member of the AIEE, the Society for Social Responsibility in Science, Sigma Xi, and Tau Beta Pi.

❖

Noah S. Prywes (M'55) was born on November 28, 1925. He received the B.Sc. degree in electrical engineering from The Technion, Israel Institute of Technology, Haifa, in 1949. From 1948 to 1950, he served in the Israeli Navy. He received the M.Sc. degree in electrical engineering from Carnegie Institute of Technology, Pittsburgh, Pa., in 1951 and the Ph.D. degree in applied physics from Harvard University, Cambridge, Mass., in 1954.



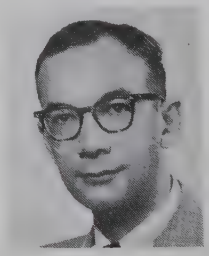
N. S. Prywes

He worked from 1954 to 1956 at RCA, Camden, N. J., as a Development Engineer engaged in computing systems utilizing magnetics and transistors. From January, 1956, to September, 1958, he was a department manager on the LARC project with Remington Rand UNIVAC, Philadelphia, Pa. In this capacity, he was responsible for all high-speed circuit design, interunit transmission, power supplies and control consoles for the LARC system. Later, he was responsible for the design, development, construction and test of the computing unit of the LARC system. Since September, 1948, he has been Assistant Professor of electrical engineering at the University of Pennsylvania, Philadelphia, retaining a consulting position with Remington Rand UNIVAC.

Dr. Prywes is a member of AIEE and Sigma Xi.

❖

Charles W. Rosenthal (S'49–A'52–M'56) was born in New York, N. Y., in October, 1928. He received the B.E.E. degree from the College of the City of New York in 1950 and the M.S. degree from Harvard University, Cambridge, Mass., in 1951. At present, he is a candidate for the Ph.D. degree at Columbia University, New York, N. Y.



C. W. Rosenthal

In 1951, he joined the Bell Telephone Laboratories, Murray, Hill, N. J., where he has been concerned with analysis and systems and logic design for military control systems and data processing systems. In 1953, he completed the Bell Telephone Laboratories Communications Development Training Program. At present, he is working on the assembly,

simulation and design automation programs needed in the development of an electronic private branch exchange.

Mr. Rosenthal is a member of the ACM, Tau Beta Pi, and Eta Kappa Nu.

❖

James A. Ruff (S'54–A'55) was born in Schuylkill Haven, Pa., on May 13, 1928. He received the B.E.E. degree from Rensselaer Polytechnic Institute, Troy, N. Y., in 1950.

J. A. Ruff

From 1950 to 1953, he served with the U. S. Navy. From 1953 to 1954, he did graduate work at Pennsylvania State University, University Park. Since 1954, he has worked for Bell Telephone Laboratories, Murray Hill, N. J., first in the Electronic Switching-System Development Department, and now in the Solid-State Device Development Department.

❖

John Schwarz (M'59) was born in Louisville, Ky., on July 28, 1927. He received the B.E.E. degree in 1950 from the Speed Scientific School of the University of Louisville, and the M.E.S. degree in applied physics in 1952 from Harvard University, Cambridge, Mass.

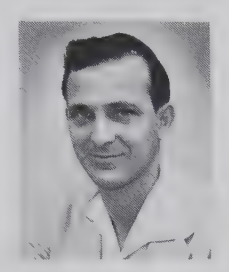
J. Schwarz

After three years at Philco Corporation, Philadelphia, Pa., where he was engaged in basic transistor development and radar pulse circuitry, he joined the UNIVAC Division of Sperry Rand Corporation, Philadelphia, in 1956. He was responsible for the design and development of a number of logic and memory circuits used in the LARC computer, and recently has been engaged in the design of advanced memory systems while also serving as a consultant to other circuit groups.

❖

Paul L. Simmons was born in Hutchinson, Kan., on February 12, 1929. He received the B.A. degree in 1953 from the University of Denver, Denver, Colo. After serving two years in the U. S. Army, 1953–1955, he earned the M.S. degree in library science from the University of Denver in 1956.

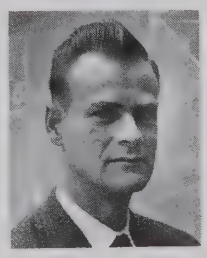
He served as Serials Librarian at the Long Beach State College Library, Long Beach, Calif., from 1956–1958. In October, 1958, he joined the System Development Corporation, Santa Monica, Calif., where he is presently Reference Librarian. Mr. Simmons has made bibliographic searches on Soviet Aircraft and missiles, mechanical translation, computers in psychology, Soviet computers, computers in medicine, oceanography, information processing in command and control systems, and satellite surveillance and tracking systems in connection with corporate projects.



P. L. Simmons

❖

Robert F. Simmons was born in Quincy, Mass., on May 14, 1925. He received the B.A., M.A., and Ph.D. degrees in psychology in 1949, 1950, and 1954, from the University of Southern California, Los Angeles.



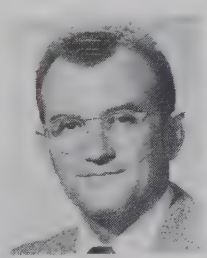
R. F. Simmons

After his graduate work, he spent two years with the Douglas Aircraft Corporation, Santa Monica, Calif., developing computerized methods of statistical forecasting of cost information. For the past five years, he has been employed at the RAND Corporation and its recent offshoot, the System Development Corporation, Santa Monica, where he is doing research toward the construction of a general-purpose cognitive language processer, to be called Synthex.

Dr. Simmons is a member of the American Psychological Association, of Sigma Xi, and is a Certified Psychologist in California. He has taught at the University of Southern California and, on occasion, has acted as psychological consultant to various research projects.

❖

Charles H. Single (A'53–M'58) was born in Chicago, Ill., on June 3, 1926. He received the B.S.E.E. degree in 1949, and the M.S. degree in 1950, both from Michigan State University, East Lansing.



C. H. Single

From 1950 to 1951, he was a graduate teaching assistant at Michigan State. From 1951 to 1955, he was employed as a Senior Scientist at the Westinghouse Electric Corporation, Atomic Power Division, Pittsburgh, Pa. Currently, he is Manager of the Computer Engineering Department, Berkeley Division, Beckman Instruments, Inc., Richmond, Calif

Mr. Single is a member of ACM, Tau Beta Pi, Phi Kappa Phi, and Pi Mu Epsilon.

❖

Henry L. Stadler was born in Columbia, Mo., on March 28, 1923. He received the A.B. degree from Harvard University, Cambridge, Mass., in 1948, and the S.M. and Ph.D. degrees from the University of Chicago, Chicago, Ill., in 1951 and 1954, respectively.

H. L. Stadler

From 1954 to 1960, he was a member of the technical staff of Bell Telephone Laboratories, Murray Hill, N. J., engaged in the development of barium-titanate ferroelectric devices and the permanent magnet-twistor memory. Since 1960, he has been with the Ford Motor Engineering and Research Center, Dearborn, Mich.

Dr. Stadler is a member of the American Physical Society, Sigma Xi, and Phi Beta Kappa.

❖

Joseph Tierney (S'54–M'57) was born in New York, N. Y., on January 14, 1934. He received the B.S.E.E. and M.S.E.E. degrees from the Massachusetts Institute of Technology, Cambridge, in 1956. While at M.I.T., he was engaged in the cooperative course program with assignments at the Bell Telephone Laboratories. During his last year, he held a research assistantship at the Research Laboratory of Electronics at M.I.T.

J. Tierney

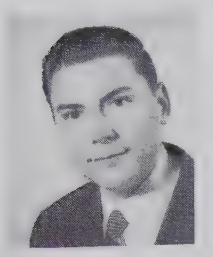
Upon graduation, he joined the technical staff of Hermes Electronics Company, Cambridge (formerly Hycon Eastern, Inc.) and was assigned to the Digital System Group, where he was engaged in computer reliability studies. In 1957, he was assigned to the communications group of the same company to work on the design of the ACE HIGH tropospheric communication network for NATO, Paris, France. On his return to the U. S. in 1958, he continued in system design of digital communication systems. In 1960, he joined the Communications and Data Processing Operation at Raytheon Company, Waltham, Mass., where he is engaged in radar pulse compressions, communication system design, and related problems.

Mr. Tierney is a member of Tau Beta Pi, Eta Kappa Nu, and SIAM.

❖

Reuben Wasserman (S'53–A'54–M'59) was born in New York, N. Y., on June 5, 1929. He received the B.E.E. degree from

the College of the City of New York in 1953, and the M.S.E.E. degree from The University of Michigan, Ann Arbor, in 1956.



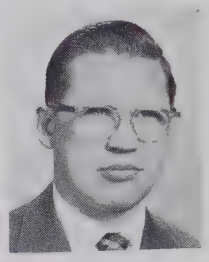
R. WASSERMAN

From 1948 to 1951, he was a sales engineer for Goodwin and McCall Company (Manufacturer's Representative). From 1953 to 1956, he was a graduate research assistant at the Willow Run Laboratories, The University of Michigan, engaged in digital computer design and circuitry. He also worked on the designs of a magnetic-core storage system, diode-capacitor storage system, and an acoustic-delay line storage. He joined Hermes Electronics Company, Cambridge, Mass. (formerly Hycon Eastern, Inc.), in 1956 as Project Engineer, responsible for the design and development of a series of products for magnetic tape data acquisition systems. He also worked on computer reliability studies and was responsible for developing redundant digital building blocks. He is presently Group Leader of the Digital Equipment Group, where he is responsible for supervising research-development on digital devices, design of data-processing systems, and development of special digital products.

He was a faculty member in the evening division of Northeastern University, Boston, Mass., from 1956 to 1958, where he taught introductory courses in electromagnetic wave theory and transistor circuit theory.

Mr. Wasserman is a member of Eta Kappa Nu, Tau Beta Pi, Sigma Xi, and has a New York state professional engineer's license.

Robert K. York (M'59) was born in Buffalo, N. Y., on December 23, 1930. He received the B.E.E. degree from Rensselaer Polytechnic Institute, Troy, N. Y., in 1957.



R. K. YORK

In 1957, he joined the Bell Telephone Laboratories, Murray Hill, N. J., where he has worked on ultrasonics, magnetic and semiconductor pulse and digital circuitry. At present, he is engaged in the development of semi-permanent memories in the Special Systems Development Department at Bell Laboratories.

Mr. York is a member of Tau Beta Pi, Eta Kappa Nu, and Sigma Xi.

Reviews of Books and Papers in the Computer Field

E. J. McCluskey, Jr., reviews editor

T. C. Bartee, J. S. Bomba, W. J. Cadden, M. Lewin, and A. A. Mullin
assistant reviews editors

Please address your comments and suggestions to the Reviews Editor, Prof. E. J. McCluskey, Jr., Department of Electrical Engineering, Princeton University, Princeton, N. J.

A. Combinational Switching Circuit Theory and Boolean Algebra

R61-84 Boolean Algebra and Its Applications—J. Eldon Whitesitt. (Addison-Wesley Publishing Co., Inc., Reading, Mass.; 1961. 182 pages.)

The seven chapters of this introductory text contain a concise and unified treatment of the applications of Boolean algebra to symbolic logic, probability, and switching circuits. Emphasis is placed upon the inherent similarities to be found in these fields through the use of a consistent algebraic notation in all chapters. This text was written for use in a one-semester course for mathematics and engineering students and for other readers with limited formal training in mathematics. The author's intention has been to provide background and motivation for further study of any of the aforementioned applications of Boolean algebra.

This objective has certainly been attained. Consequently, readers should not be disappointed to discover that the treatment of the algebra itself and that of the related fields are not complete within this treatise. As encouragement for the reader to satisfy a further desire for more exhaustive coverage of the subject matter, the author has mentioned appropriate references and pointed out limitations of the methods discussed.

The first two chapters are devoted to an intuitive introduction to the algebra of sets and a formal development of definitions and theorems of a Boolean algebra. In the second chapter, the author admits to some repetition of introductory material in the interest of formalization. Chapter three contains a discussion of symbolic logic and lucid treatments of material implication and the role of Boolean algebra as an aid to reasoning. The application of Boolean algebra to contact switching circuits is presented in chapter four. It is surprising to find that non-series-parallel, multiterminal, and symmetric circuits are all discussed in this chapter, which, *in toto*, occupies the space of only twenty-eight pages. The concise treatment is made possible by the author's justified refusal to ramify the discussion in applications in which Boolean algebra is not particularly useful. Sequential relay circuits are next discussed in chapter five from the standpoint of *operate* and *hold* paths for relay control and sequence diagrams for circuit synthesis. Chapter six contains a very brief discussion of number systems, flip-flops, and examples of logic used in performing most of the operations of binary arithmetic with positive numbers. The application of Boolean algebra to the subject of probability is then introduced in chapter seven.

The unique contribution represented by this book is the unified treatment of the various applications of Boolean algebra. Professor Whitesitt has used a concise and direct style which is very well suited to an introductory text. Interesting and appropriate examples and exercises are provided at the end of most sections. Since the problems are not difficult and the solutions of many are provided, the book would be useful for self-study as well as for classroom application.

The author has shown remarkable restraint in his treatment of the subject and in maintaining uniformity of conceptual level in his text. For example, he has chosen switching-circuit synthesis methods to illustrate in the most direct way the applications and implications of the algebra rather than more general techniques which would be employed more often in professional design work. Although this specific observation is not made in the text, the author does point out that many practical problems are not treated. These problems are those in which Boolean algebra has not been successfully applied.

Only relatively minor criticisms of this book can be offered. It is felt by this reviewer that the coverage of Boolean algebra theorems might have been slightly more complete to include at least the development of the highly useful property

$$xy + yz + x'z = xy + x'z.$$

The more extensive treatment of switching circuits relative to other applications is considered to be justified in view of interest and importance. However, it seems that the treatment of probability should have been given more emphasis. The exercises are uniformly relevant and instructive, but it is believed that more difficult problems should have been included to provide a greater challenge to college students. The several errors which were noted in reading the book are relatively obvious and should not detract in any way from the effectiveness of the text.

Dan H. Daggett
General Dynamics, Fort Worth, Texas
A Division of General Dynamics Corp.

R61-85

1) Linear Input Logic—Robert C. Minnick. (IRE Trans. on Electronic Computers, vol. EC-10, pp. 6–16; March, 1961.)

2) Unate Truth Functions—Robert McNaughton. (IRE Trans. on Electronic Computers, vol. EC-10, pp. 1–6; March, 1961.)

3) Axiomatic Majority-Decision Logic—M. Cohn and R. Lindaman. (IRE Trans. on Electronic Computers, vol. EC-10, pp. 17–21; March, 1961.)

4) Arbitrary Boolean Functions of *N* Variables Realizable in Terms of Threshold Devices—O. B. Stram. (Proc. IRE, vol. 49, pp. 210–220; January, 1961.)

Largely due to the advent of the parametron, interest in threshold device logic, originally centered on magnetic cores, has been revived. The general problem being attacked with renewed ferocity is: given a Boolean function, how does one realize this function with threshold devices?

Probably the most fundamental attack on this problem starts by trying to devise simple tests for Boolean functions which can be realized with a single threshold device. A test is immediately implicit in the definition of a threshold device. A threshold device has n inputs: $p_1, \cdots, p_i, \cdots, p_n$; and an output ϕ (p_i and ϕ take one of two values: 0 or 1). The output ϕ is 1 if, and only if:

$$x_T + \sum_1^n x_i p_i \geq 0$$

where x_i are rational numbers, and $+$ is the ordinary addition sign.

The condition that a given Boolean function $f(p_1, \cdots, p_n)$ should be realized by a single threshold device, results in a set of inequalities:

$$x_T + \sum_{i=1}^{n} x_i p_{ij} \geq 0 \qquad \text{for all } j \text{ such that } f(p_{1j} \cdots p_{ij} \cdots p_{nj}) = 1$$
$$x_T + \sum_{i=1}^{n} x_i p_{ik} < 0 \qquad \text{for all } k \text{ such that } f(p_{1k} \cdots p_{ik} \cdots p_{nk}) = 0. \tag{1}$$

The question then is: is there a set of x's which satisfy this set of inequalities? Both McNaughton and Minnick develop this approach to the problem.

McNaughton makes use of the known fact that the function $f(p_1, \cdots, p_n)$ must be unate (representable in a form in which no variable appears both negated and unnegated) in order for (1) to have a solution. In addition, he notes that the set (1) may be written, j and k being the same as in (1), as:

$$\sum x_i p_{ij} \geq -x_T > x_i p_{ik} \tag{a}$$
$$\sum x_i p_{ij} > \sum x_i p_{ik} \tag{f}$$

where the solution of (a) is possible if, and only if, (f) has a solution.

Although this leads to more equations than the first set of equations, a variable x_T has been eliminated. Many of these equations are often easily seen to contradict or to imply others, thus either revealing that the function is not realizable or resulting in fewer inequalities. McNaughton's suggestion is actually one step of a general technique (elimination of variables) for solving a set of inequalities. If the further steps of this procedure were carried out, we would have a complete procedure for solving.[1]

Minnick, on the other hand, suggests that one apply the Simplex Method to solve the set (1). Minnick uses this approach as part of an attack on the more general problem of designing multidevice threshold logic. However, beyond the single threshold case, he guarantees very little about the minimality of the circuits derived. With this technique and various heuristic devices, and the assistance of friends, he has completed a very useful table of minimal threshold circuits for all four (two-stage) variable functions.

A less fundamental but simpler approach to designing logical circuits with threshold devices is to take advantage of special Boolean functions which are "natural" to threshold devices. Minnick uses the fact that a single threshold device can be used to realize one of two "fundamental functions"—an "and" function or an "or" function (each including primed variables). These are obviously sufficient to realize any Boolean function. Minnick, in fact, gives four different ways of converting a Boolean function into two-stage combinations of fundamental functions.

Cohn and Lindaman, on the other hand, make use of the fact that a single threshold device can be used to realize the "majority function"—Majority $(x, y, z) = xy + xz + yz$. They show that Majority $(x, y, 0) = xy$, and Majority $(x, y, 1) = x + y$; and therefore, that any Boolean function can be realized with majority functions (given primed literals). In addition, Cohn and Lindaman give a set of axioms and derive a set of theorems for manipulating these majority functions and for converting from Boolean to majority function expressions. In fact, Cohn and Lindaman, contrary to their stated limitations to majority elements, include inverting elements. This is seen in the circuit they derive for a one-stage adder. This is unnecessary, but their example provides some simplification, and perhaps some confusion.

Stram shows how one may build threshold functions using two variable threshold functions as basic units. His approach is essentially enumerative and seems exceedingly laborious. This laborious approach can in theory be extended so as to use n-variable basic build ing units. It was difficult to extract the contribution from this article.

There are obviously many sets of fundamental functions which could be used as a basis for design of threshold circuits. It seems that more important than setting forth such new sets would be the development of techniques for evaluating them.

In addition to providing general methods for designing threshold logic, special methods may usefully be developed for special classes of logic circuits. For symmetric (k out of n) functions, Minnick presents a very nice technique for designing threshold circuits. The result is the simplest threshold circuitry for such functions which has yet been published.

M. C. Paull
Bell Telephone Labs., Inc.
Whippany, N. J.

[1] A. W. Kuhn, "Solvability and consistency for linear equations and inequalities," *Am. Math. Monthly*, vol. 63, pp. 217–232; April, 1956.

R61-86

1) Reti di commutazione constituite con circuiti a soglia (Switching Networks Built from Threshold Circuits)—Luigi Dadda. (*L'Elettrotecnica*, vol. 47, pp. 1–8; 1960.)

2) La sintesi delle reti di commutazione composte da circuiti a soglia (The Synthesis of Switching Networks Built from Threshold Circuits)—Luigi Dadda. (*Alta Frequenza*, vol. 30, pp. 127–134; February, 1961.)

These two Italian-language papers are concerned with the problem of synthesizing binary functions by means of threshold devices. The problem is treated in a systematic general way, so that results apply to the case of such circuits as magnetic-core circuits, resistor-transistor or resistor-tunnel-diode circuits, parametrons, etc.

In the first paper, only the case of symmetric functions is treated. In the second, such restriction has been relaxed and arbitrary functions are considered. To clarify at once the approach taken by the author, it must be said that no recourse has been made to logical primitives in the Boolean sense (such as AND, OR, NOT, etc.), rather binary functions are grouped into classes and a "prototype" threshold expression is sought for each class.

In the simpler case of symmetric functions of n variables, the following expression is assumed:

$$U = S_s[A, p_1 S_{s_1}(A), p_2 S_{s_2}(A), \cdots, p_m S_{s_m}(A)]$$

being $s, s_1, s_2, \cdots, s_m$ threshold indexes, $p_1, \cdots, p_m$ the weights of the inputs to the S_s threshold device, and A being the set of the n variables $A_1, \cdots, A_n$.

It has to be noted that few other assumptions are implicit in that expression; the variables are all taken in their unprimed or primed form and with weight equal to one; and two-level circuitry is assumed. The author shows that any n-variable symmetric function can be synthesized using that expression, and a procedure is indicated to determine $s, p_1, s_1, \cdots, p_m, s_m$, given the a-numbers of the symmetric function for different values of m. The expressions with the maximum possible number of sections $C_{\max} = 2m + 2$ are taken as prototype expressions from which the cases for $C < C_{\max}$ can be derived.

In the second paper, the more general problem of arbitrary Boolean functions is considered. As anticipated, functions are grouped into classes, and a prototype expression is again sought for each class from which all the other expressions relative to functions of the same class can be obtained for any permutation of the variables and for any choice of their form (primed or unprimed). The greater complexity of the problem requires one to consider weighted literals (A^*) instead of only unit-weighted, all unprimed, or primed variables.

Prototype expressions are sought first among elementary threshold functions (1) or, if this is not possible, among complex threshold functions (2) and (3):

$$F = S_s(p_1 A_1^*, p_2 A_2^*, \cdots, p_n A_n^*) \tag{1}$$
$$F = S_s(p_1 A_1^*, \cdots, p_n A_n^*; p_1' S_{s_1}; \cdots p^{(m)} S_{s_m}) \tag{2}$$
$$F = S_s(p' S_{s_1}; p'' S_{s_2}; \cdots p^{(m)} S_{s_m}). \tag{3}$$

The paper shows that:

1) Only part of the 2^{2n} n-variable functions can be expressed by elementary threshold functions. These have been completely determined for $n = 2, 3, 4$.
2) The remaining nonelementary functions can all be expressed by means of (2) or (3). The author anticipates the results of a general method valid for any n-variable arbitrary function, which will appear in a later paper.
3) For $n = 2, 3$, any function can be put into form (2) using only two threshold devices.

It appears the field is still open to further investigation; the author himself outlines some problems to be solved next, as for instance, a general synthesis procedure for any choice of n and the additional possibility of determining the value of s leading to the most economical circuit.

The papers are generously provided with tables showing results and intermediate steps.

For those unfamiliar with the Italian language, it may be of some comfort to know that the second of these articles will appear in English in the March issue of *Alta Frequenza*. This issue will be the first of a series in which articles originally published in Italian will appear in English translation.

Danilo Zucoli
Olivetti—LRE
Borgolombardo
Milano, Italy

R61-87 Computer Design of Multiple-Output Logical Networks—T. C. Bartee. (IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-10, pp. 21–30; March, 1961.)

This is an excellent paper the scope of which is perhaps best indicated by quoting the author's summary:

An important step in the design of digital machines lies in the derivation of the Boolean expressions which describe the combinational logical networks in the system. Emphasis is generally placed upon deriving expressions which are minimal according to some criteria. A computer program has been prepared which automatically derives a set of minimal Boolean expressions describing a given logical network with multiple-output lines. The program accepts punched cards listing the in-out relations for the network, and then prints a list of expressions which are minimal according to a selected one of three criteria. This paper describes the basic design procedure and the criteria for minimality.

The adverse criticisms are minor and concern the title. The title does not indicate that multiple inputs as well as multiple outputs are considered. Furthermore, the use of the words " . . . Design of . . . Logical Networks" is misleading since the paper considers the minimization of combinational Boolean expressions and is not directly related to network design. One excellent feature of this paper is that fundamentals are presented clearly so that reference to other papers is not essential for understanding the techniques presented.

BERNARD HARRIS
New York University
New York, N. Y.

R61-88 Minimization of Multiple-Output Switching Circuits—R. B. Polansky. [*Trans. AIEE* (*Commun. and Electronics*, no. 53), pp. 67–73; March, 1961.]

All of the major pitfalls for the unwary have been avoided in this paper which gives a method of simultaneously minimizing a set of Boolean functions of the same variables in two levels of gating. All the usual qualifications to the sterile word "minimization" apply.

The method tags each canonic term of a function with a q-length binary number indicating to which of the q functions it belongs. These tagged terms are then combined as though one function and the Quine-McCluskey[1] method is applied with an additional rule for combining tags. A simple cost function is associated with each resulting prime implicant, and the prime implicant chart is then attached with vigor. The five steps in the method given for selecting the prime implicants leave something to be desired; certainly unwieldy for hand procedures, it required a flow chart for the author's explanation of the steps involved. Nonetheless, the author clearly did not have people in mind when setting up his technique.

The problem of generating multiple-output prime implicants by combining them into a single equation was suggested seven years ago by Muller as referenced by Polansky. Muller, however, operated on his tags as though they were Boolean variables, considerably increasing the size of the problem. Polansky correctly saw the tags should be operated on differently. Thus, Muller successfully avoided the main problem which has been to find a good way of selecting the prime implicants.

Polansky's algorithm for selecting prime implicants is sufficient but hardly necessary. The reviewer's paper[2] on this subject offers this alternative: maintaining the tags on the prime implicants by applying the McCluskey[3]-Petrick[4] method of writing a conjunctive equation of the prime implicants disjunctively implying each canonic term. Minimizing this equation with the rule $A \cdot (A+B) = A$ yields all *irredundant* solutions to which cost functions may be assigned for selecting the *minimal* one. It appears, however, that this method would have much of its burden reduced if a few simple observations, such as Polansky's first few steps, are first applied.

It is also worth observing with Roth that simplifying each function separately and then taking the intersections of the prime implicants has something in favor of it as a time-reducing method of getting the multi-output prime implicants.

One thing ignored by most papers on this subject (including the recent one by Bartee),[5] is that the costs involved change considerably when going from one- to many-output minimization. Namely, the question from the real world, whether the gate being shared among the various outputs has the drive necessary, and what cost is required to give it the necessary drive. This question is crucial to the whole multi-output notion of sharing gates and is a particularly painful one when the authors use diodes as their example—and they always do.

Nonetheless, it is comforting to find the multiple-output problem handled adequately at last, even though it is nine years since Quine published his solution for single outputs.

JOHN EARLE
IBM Corp.
Poughkeepsie, N. Y.

[1] E. J. McCluskey, Jr. "Minimization of Boolean functions," *Bell Sys. Tech. J.*, vol. 35, pp. 1417–1444; November, 1956.
[2] G. A. Maley and J. Earle, "Synthesizing multiple-output switching networks," IBM Tech. Rept. 00.733.
[3] McCluskey, *op. cit.*, Section 9.
[4] S. R. Petrick, "A Direct Determination of the Irredundant Forms of a Boolean Function from the Set of Prime Implicants," AF Cambridge Res. Ctr., Bedford, Mass., Tech. Rept. No. AFCRC-TR-56-110; April, 1956.
[5] See Review R61-87.

R61-89 An Algorithm for Determining Minimal Normal Forms of an Incomplete Truth Function—Thomas J. Mott, Jr. [*Trans. AIEE* (*Commun. and Electronics*, no. 53), pp. 73–76; March, 1961.]

The author considers the following problem: given a set of input configurations I for which an output must be one, and a set of input configurations D for which the output may be defined arbitrarily, find all the disjunctions F_i of subsets of the prime implicants of $I \vee D$ such that $I \rightarrow F_i$ and, further, each F_i has the property that, if any term of F_i is deleted, the implication no longer holds. Such F_i are defined as weakly irredundant normal weak equivalents of I. The author presents a method for producing all such equivalents.

This reviewer feels that the problem is of academic interest only. Moreover, he feels the following method is simpler yet more efficient:

1) Find all the prime implicants of $I \vee D$.
2) Using Gazale's ratio test,[1] compare the list of prime implicants against each term of any normal equivalent of I. Each term will produce one or more irredundant sets of prime implicants whose disjunction is implied by that term.
3) Combine these implications as described by both Mott and Gazale to derive the desired F_i.

Theorems 2 and 3 could be stated more strongly since they are true for each term of any alternational normal form equivalent of I and not just the developed normal equivalent of I.

It may be of interest to note that the IBM 704 minimization program based on the work of Roth[2] presently produces one such equivalent and could easily be modified to produce all such equivalents.

W. W. BOYLE
IBM Corp.
Poughkeepsie, N. Y.

[1] M. J. Gazale (Ghazala), "Irredundant disjunctive and conjunctive forms of a Boolean function," *IBM J. Res. & Dev.*, vol. 1, pp. 171–176; April, 1957.
[2] A. E. Randlev, "Computation of a Minimum Two-Level And-Or Switching Circuit," SHARE, PKMIN2: April 22, 1959.

R61-90 A Visual Matrix Method for Minimizing Boolean Functions—A. D. Zakrevskii. (*Automation and Remote Control*, vol. 21, pp. 255–258, November, 1960; translated from *Avtomatika i Telemekhanika*, vol. 21, pp. 369–373; March, 1960.)

This paper presents a "semi-algorithmic, semi-intuitive" method for using the standard Karnaugh map to handle the minimization of switching functions of more than four variables. Its principal contribution is a clear and simple method for determining those combinations of map elements which constitute subcubes of the function represented on the map.

As in the standard Karnaugh method, a symmetric Gray code is used to represent the Boolean function on the map. The variables are divided into two sets, as equal in size as possible, and distributed along the rows and columns, respectively, of the matrix. Vertical and horizontal "zones of symmetry" of width 2^r are next defined on the map, each zone being divided by "an axis of symmetry of rank r." For example, an 8×8 map will have 8 zones of symmetry of width 2 (rows 1 and 2, 3 and 4, 5 and 6, 7 and 8, and columns 1 and 2, 3 and 4, 5 and 6, 7 and 8, respectively), 4 zones of symmetry of width 4 (rows 1 to 4, and 5 to 8, and columns 1 to 4, and 5 to 8), and 2 zones of symmetry of width 8. The concept of an "interval of rank r," corresponding to the conjunction of r variables in the disjunctive canonical form or to a subcube of 2^{n-r} elements, is then introduced. Starting with intervals of rank n equivalent to single-cell matrix elements, it is now possible to form intervals of successively smaller rank as follows: *a set of 2 intervals of rank k forms an interval of rank k−1 if the intervals are symmetric with respect to some axis of symmetry and are found completely within its zone of symmetry.*

All intervals forming partial covers are isolated visually, and a cover of the complete map is chosen in the usual way. Both minimal disjunctive and conjunctive forms of switching functions can be determined, and "don't care" conditions are easily accommodated. To find, for example, the minimum disjunctive form of a switching function, the map must be decomposed into intervals in such a way that the sum of the ranks is minimal.

The author claims that it is easy to notice the symmetry visually, and this reviewer would tend to agree. It is also claimed that functions of up to ten to twelve variables can be handled with a little practice. The author is less hopeful concerning the possible mechanization of the process.

The presentation is generally clear, except for some unusual terminology due undoubtedly to the translation process. In particular, functions which include "don't care" conditions are termed "functions which differ in their values on certain sets of variables," and two adjacent elements are rendered as "two series of elements." Moreover, the equation on line 7 of page 258 should read

$$f = (a + bc + de)(b + \bar{a} + \bar{c}d).$$

To summarize, the observations contained in this short paper are implicit in other descriptions of the use of Karnaugh maps. However, the symmetric properties of subcubes useful in their visual determination have not, to the reviewer's knowledge, been stated so clearly before. As such, the paper constitutes a worthwhile contribution to the minimization field.

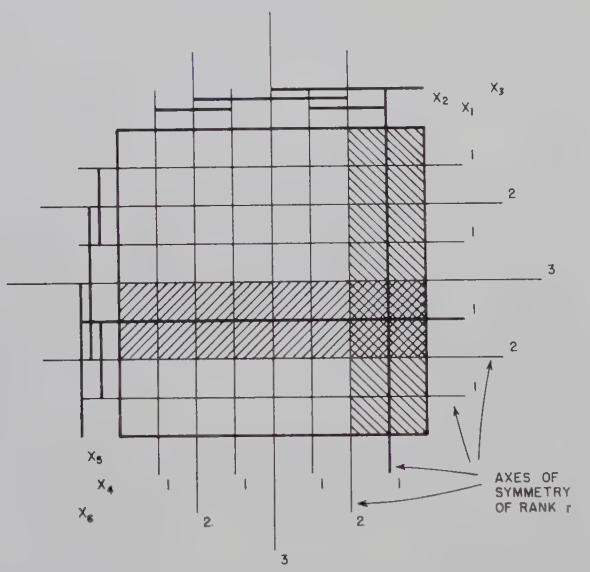


Fig. 1—Six-variable Karnaugh map exhibiting all axes of symmetry, and two selected zones of symmetry of width 2.

Gerard Salton
Harvard University
Cambridge, Mass.

B. Sequential Switching Circuit Theory and Iterative Circuits

R61-91 Circuit Synthesis by Solving Sequential Boolean Equations—Hao Wang. (*Z. math. Logik und Grundlagen d. Math.*, vol. 5, pp. 291–322; 1959.)

The author presents an extremely interesting, thorough investigation of the problems involved in obtaining a set of realizable explicit functions from an implicit Boolean equation $H(i, \cdots, j, x, \cdots, y, di, \cdots, dj, dx, \cdots, dy) = 0$. In this equation $i, \cdots, j$ are the inputs, and $x, \cdots, y$ are the outputs; the operator d is such that di and dx, for example, are the next subsequent values of i and x (contrary to popular usage of d or D as a delay operator). Solutions are sought for the outputs $dx, \cdots, dy$.

The first problem is to determine whether the outputs $dx, \cdots, dy$ in H can be expressed as Boolean functions of the inputs $di, \cdots, dj$, the previous inputs $i, \cdots, j$, and the previous outputs $x, \cdots, y$, and, if so, to obtain the appropriate expressions called *deterministic sequential functionals*. Effective procedures for both of these tasks are given. Incidentally, the author purposely avoids making a distinction between output states and internal states. Considering the output states as internal states, one sees that his deterministic functionals are essentially equivalent to the sequential machines of Moore.[1]

A second problem arises when there exist effective solutions for $dx, \cdots, dy$ which cannot, however, be expressed as deterministic functionals. These solutions are called *predictive sequential functionals* and are typically functions of $i, \cdots, j, di, \cdots, dj, d^2i, \cdots, d^2j, \cdots, d^ni, \cdots, d^nj, x, \cdots, y$. Again, effective procedures are given for determining whether the outputs $dx, \cdots, dy$ in H can be so expressed, and, if so, procedures are given for obtaining the expressions. Functionals of both the deterministic and predictive types are seen to lead to circuit realizations; in the latter case, the author notes that a realization may be obtained essentially by reducing it to the deterministic case. Intuitively, this means that one must wait until the last input involved is obtained before producing the output.

Finally, there exist equations H for which there is no effective solution for $dx, \cdots, dy$ in terms of predictive functionals, but for which there is nonetheless a solution. For such cases an effective procedure is shown for obtaining a *solution table* which characterizes the solution as well as guaranteeing its existence. In these cases, a circuit realizaton is not possible.

Several other points are worth mentioning. The author shows that every set of sequential Boolean equations is reducible to one equation H. Furthermore, the procedures involved in solving H for $dx, \cdots, dy$ usually lead not to single solutions, but rather to general solutions which contain free parameters. Specialization of these parameters leads to solutions of varying complexity with correspondingly varying initial output values $x_0, \cdots, y_0$; the simplest solution is thus desired for the realization, as long as its initial output values are compatible with the desired behavior. Incidentally, the initial output values also play a role in the noneffective solutions. It is because these values may depend on the entire infinite future that the solutions are in fact noneffective.

Various related questions are raised by the author, some of which remain unanswered. The problems of simplification and of solvability subject to input restrictions, for example, are mentioned only in passing. The notions of adding to the sequential Boolean equations restricted quantifiers, certain modulo operators and time operators other than d are discussed in some detail. Turing machines are considered in this context, and an alternate proof of Kleene's normal form theorem is given. In all, the paper is well organized, well written and informative.

Peter G. Neumann
Bell Telephone Labs, Inc.
Murray Hill, N. J.

[1] See [13] of Wang.

R61-92 Finite Automata—M. A. Aizerman, L. A. Gusev, L. I. Rozoner, I. M. Smirnova, and A. A. Tal'. (*Automation and Remote Control*, "Part 1," vol. 21, pp. 156–164, October; "Part 2," vol. 21, pp. 248–255, November; 1960.)

Most of this paper is a clearly written, carefully translated survey of some results in the theory of finite automata. The term "finite automaton" is here used as a synonym for "sequential machine," "sequential circuit," or "logical network," and the paper brings together and connects some of the work which has been done in these areas during the past few years, discursively treating material from many sources. Readers familiar with the work of Kleene, Huffman, McCulloch and Pitts, Burks, Moore, Wang, etc., will find this paper a pleasant treatment of familiar material; the uninitiated should find this a straightforward (nonrigorous) introduction.

The discussion ranges through state tables, equivalent machines, graphs and state diagrams, a little material on neuron networks, and finally relay circuits, touching lightly and referencing further material on each subject. (The references indicate that most of the major English-language papers in these areas have now been translated into Russian.) There are a few none-too-clear paragraphs which worry about finite-state machines with infinite tapes (without mentioning the name *Turing*) and algorithms for discovering "contradictory triads" on infinite tapes. Having noted that infinite tapes cannot be "given," the authors might well have passed quickly on.

T. C. Bartee
Lincoln Lab., Mass. Inst. Tech.
Lexington, Mass.

C. Pattern Recognition and Learning Theory

R61-93 What Is an Intelligent Machine?—W. Ross Ashby. (*Proc. Western Joint Computer Conf.*, Los Angeles, Calif., May 9–11, 1961; pp. 275–280.)

This paper by Professor Ashby consists primarily of an elaboration of two broad statements: "Not one clear counter-example has been given in the last ten years to show that an intelligent system is anything other than one which achieves appropriate selection," and "Any system that achieves appropriate selection (to a degree other than chance) does so as a consequence of information received." These statements, at least with a sympathetic interpretation, are unexceptionable. They even sum up a certain approach to problems of intelligence in the same sense that "morphogenesis is nothing but a sequence of integrated biochemical reactions" tell us of one approach to the problems of morphogenesis.

By themselves, however, the statements convey little beyond an indication of viewpoint; for any deeper meaning, both depend in an essential way upon certain undefined terms. For example, the word "appropriate" in the phrase "appropriate selection" vitiates both sentences unless one has a criterion for distinguishing appropriate selections from inappropriate ones. The criterion must be both precise and of a sufficient generality to merit consideration at the same level of generality as the word "intelligence." If a different *ad hoc* criterion of appropriateness must be set up for each new case of intelligent action encountered, then the general statement degenerates into a collection of special statements and, in a great measure, both generality and usefulness are lost. In particular, a criterion for distinguishing appropriate selections requires some discussion of the means of selection, that is, selection operators, the arguments or inputs of the selection operators, and the possible alternatives or range of selection. At least for the reviewer, Ashby's remarks here (and his brief examples) do not go far enough to give implicit definition to his ideas of "appropriateness," and nowhere in the paper does he give an explicit discussion of these difficulties. In order even to begin to consider evidence for or against the first statement quoted (let alone try to formulate a proof or a counter-example), some such discussion is a necessity.

Although it is difficult in a short expository paper, it would have been a service to the reader if Ashby had at least indicated the nature of some of the deeper (and more formal) problems which occur in any attempt to employ the quoted statements in the study of "intelligent machines."

John H. Holland
University of Michigan
Ann Arbor, Mich.

D. Digital Computer Systems

R61-94 High-Speed Arithmetic in Binary Computers—O. L. MacSorley. (Proc. IRE, vol. 49, pp. 67–91; January, 1961.)

This paper compares several logical designs for a parallel binary high-speed arithmetic unit. The name "report" is well chosen by the author because the methods of high-speed addition, multiplication and division are presented in exhaustive detail. The viewpoint is entirely practical; equations are avoided and detailed descriptions and rules of operation are stated in words. A typical set of logical circuits is assumed and used to compare the relative speeds and equipment cost of alternate high-speed designs. Binary addition (including subtraction by use of two's complement) multiplication and division are considered in separate sections of the paper.

The first section describes logical designs of two well-known types of parallel adders—the simultaneous-carry adder[1] (classified as fixed-time adder) and the carry-completion sensing adder[2] (classified as variable-time adder). The designs are compared for a 100-bit word length. The simultaneous-carry adder is further considered for three intermediate variations between serial carry propagation and complete carry look-ahead; all five versions are compared for word lengths of 50 and 100 bits.

The section on multiplication discusses methods which minimize the number of additions by means of multiplier recoding and the application of stored-carry (carry-save) adders. Complete rules with numerical examples are given for multiplier recoding with variable-length shifting and with fixed-length shifting of two and three binary positions. The discussion of carry-save adders describes an ingenious cascading of three carry-save adders and one conventional adder into a system which handles eight multiplier bits at once. Simplifications of carry-save adders which lead to component reduction are thoroughly explored, and the result is an interesting and novel high-speed multiplication system.

Three high-speed variations of nonrestoring division are discussed in the last section. Initially normalized divisors and variable-length shifting are assumed. In all cases, the partial remainders are normalized to achieve high-speed operation. Different multiples of the divisor are available for the formation of the next partial remainder: one times the divisor in the first case, one-half, one, and two times the divisor in the second case. Exhaustive tables of shift lengths are constructed for all allowed five-bit divisors and dividends. An evaluation of these tables leads to the interesting third case: the use of three-fourths, one, and three-halves times the divisor as the practically optimum set of multiples. A comparison of eight distinct versions of division—two variations each of the first two cases and four variations of the third case—concludes the discussion. The verbal description of nonrestoring division with normalization of partial remainders (called "shifting across zeros and ones") is rather awkward. The reader must deduce the representation of negative numbers, especially the rules for the handling of negative divisors. Particularly, the term "addition" (page 81, left column, line 23) is used for "subtraction" or "addition of two's complement" of the divisor. It is unfortunate that four years were needed for this version of accelerated nonrestoring division[3] (credited to a February, 1957, internal paper of IBM by J. Cocke and D. W. Sweeney) to reach general publication.

The paper presents a wealth of detailed information, highlighted by its contributions of the method for efficient cascading of carry-save adders in multiplication and the systematic tabular investigation of most useful divisor multiples in division. On the debit side, the theoretical treatment of arithmetic and the details of logical design for specific cases are not sufficiently distinguished, and the long verbal statements of rules and procedures lack the conciseness of mathematical formulation. The bibliography is given at the end and does not directly identify the sources of various methods discussed in the text. Although these matters will not daunt the expert, they make the paper unnecessarily difficult for the generally interested reader.

Algirdas Avizienis
Jet Propulsion Lab.
California Institute of Technology
Pasadena, Calif.

[1] A. Weinberger and J. L. Smith, "A one-microsecond adder using one-megacycle circuitry," IRE Trans. on Electronic Computers, vol. EC-5, pp. 65–73; June, 1956.

[2] B. Gilchrist, J. H. Pomerene and S. Y. Wong, "Fast carry logic for digital computers," IRE Trans. on Electronic Computers, vol. EC-4, pp. 133–136; December, 1955.

[3] Another version has been independently described by J. E. Robertson, "On the Design of a Very High-Speed Computer," Digital Computer Lab., Univ. of Illinois, Urbana, Rept. No. 80; October, 1957, Chap. 8. Also discussed by K. D. Tocher, "Techniques of multiplication and division for automatic binary computers," *Quart. J. Mech. and Appl. Math.*, vol. 11, pt. 3, pp. 364–384; 1958.

R61-95 Computer Logic—Ivan Flores. (Prentice-Hall, Inc., Englewood Cliffs, N. J.; 1960. 458 pp. $12.00.)

This book, which is appropriately subtitled "The Functional Design of Digital Computers," fills an important gap in the computer literature. Most books in the computer field are either textbooks to teach a particular subject, such as programming or logic design, or collections of descriptions of various hardware techniques, often those used in existing machines. While such books are valuable as reference material, they provide little understanding of how all the various components and subsystems of the computer work together to form a coherent system. The present book does this quite effectively.

The first five chapters of the book introduce the system concept of a computer, its operation and application, leading to the order code description of a particular example machine, Polyvac. The question of matching the computer to the problem to be solved is considered briefly, with particular emphasis on business applications. The order code for Polyvac is built up by showing how additional commands, *e.g.*, COMPARE, or additional facilities, *e.g.*, index registers, improve the efficiency of the program for a particular problem.

Chapters six through eight are concerned with the principles of machine arithmetic, number systems, and machine language. Machine arithmetic is first discussed for decimal numbers, but in the chapter on number systems the extension of the principles to other number systems is effectively illustrated. The chapter on machine

languages describes most of the commonly used codes and very briefly considers the question of error detecting and correcting codes.

The next three chapters are concerned with logic techniques and their application to the design of a number of functional logic units, which are used in later chapters as building blocks for the computer subsystems. Following a brief explanation of the principles of symbolic logic, the Karnaugh map technique is illustrated by the design of devices such as binary adders and shift registers. The logical organization of more complicated functional units such as decimal adders, comparison circuits, coding and decoding networks and accumulators are explained in terms of the simpler building blocks and the additional logic needed to tie them together.

The next four chapters are concerned with the major subsystems of the computer. Each subsystem is built up using the simple building blocks and complex functional units derived in the preceding chapters. The arithmetic unit is built up successively from the simplest operation of serial addition to the full logic needed for multiplication and division. Discussion of memories begins with a general description of delay-line memories, including the associated logic and timing necessary for their control. Description of static memories includes a relatively detailed explanation of magnetic cores and their utilization in memory systems. Magnetic drums are also described at some length; however, there is no mention of the specialized but highly efficient ways in which drums are utilized in real-time control computers.

The chapter on the control unit explains effectively how both instructions and data words can be handled in the same memory without confusion. Control logic is developed in detail for each of the classes of operations used in Polyvac. The chapter on input and output equipment describes both the operation of the devices and the logic needed to couple them with the computer. The reader is able to to see the relation between the mechanical equipment and the structure of the computer itself. A notable omission in this chapter is the lack of information concerning input-output devices for computers which must communicate with analog equipment.

By the completion of the first fifteen chapters, the reader has been carefully conducted through the structure of all the major subsystems of a typical computer and is able to visualize how these various units operate together. The final chapter considers in detail how Polyvac would be used to solve a particular problem. Although this chapter offers little additional insight into computer logic, it gives the reader an idea of the mechanics of using a computer. Some of the procedures described would be simpler (and more realistic) if the author admitted the use of simple programming aids to provide the programmer with a more convenient language than the machine language itself.

In treating as complicated a subject as digital computer systems, there is much room for difference of opinion as to the relative emphasis on various topics. This reviewer would have liked to see more discussion of the questions of synchronous and asynchronous operation, parallel and serial organizations, and absolute value and complement number representations. In light of the present trend in computer systems, some discussion of overlap and simultaneous operations would be valuable. All in all, however, the author has succeeded in providing a book which should give a good understanding of the complex structure of a digital computer system to the reader who is willing to devote careful attention to the step-by-step development of the detailed logic organization.

R. A. Kudlich
AC Spark Plug Div.
General Motors Corp.
El Segundo, Calif.

R61-96 Trends in Design of Large Computer Systems—Charles W. Adams. (*Proc. Western Joint Computer Conf.*, Los Angeles, Calif., May 9–11, 1961; pp. 361–364.)

Computers of a particular "generation" are distinguished by major structural or logical innovations which persist through succeeding generations. The computers of the present, which Adams has discussed, will most certainly be distinguished as a generation by their provisions for the simultaneous operation and autonomous control of the several parts of the computer and its associated equipment. The major structural distinction is, of course, the use of solid-state elements throughout.

It must be expected, as the author suggests, that the effort toward computers of higher computational capability and speed will explore much further the more efficient use of logical elements as now known. When technology is able to produce computing elements in the form of large functional aggregates, parallel operation of such aggregates may be as common as the parallel operation of single digit structure is now.

The new "nonerasable" store of the Atlas computer, which is discussed at length, is not an essential concept of the Atlas system. Its principal merits are the fact that its elements are inexpensive, and that its reading operation may be shorter than that of most square-loop ferrite stores in current use. An "erasable" memory of equal speed, operated in the manner in which the "nonerasable" store is used, would certainly not be at a disadvantage. Its use in the "stored logic" system, which executes complex operations as if they were single instructions although they are really subroutines in the fixed store, does not differ greatly from the use of subroutines in any contemporary computer which has efficient linking procedures. The one level addressing system of the Atlas may be a more durable innovation, since it should surely be convenient; and it may lead to a more efficient use of drum storage.

The author mentioned that several extraordinary instruction formats have appeared in the new computers but made little descriptive comment about them. It is interesting to see that single-address type instructions tend to prevail, as they should when the effort is toward maximum efficiency and maximum packing of meaningful information.

Concerning specific instructions, the author tells us no more than that interesting forms do exist in the computers mentioned. It is appreciated by most designers that operations in which long iterative processes can be accomplished with a single instruction, or with very few instructions, can do something toward improving the speed and efficiency of the computer. However, as the author points out, these savings are not likely to be spectacular, and their evaluation in advance is most difficult.

The concept which the author calls "graceful degradation" provides that a number of identical units will be included for each kind of operation which the computer must perform, and that the computer will select an available unit when an operation is required. The computer will also recognize a unit which is failing and place it in unavailable status. This concept has been considered and discussed by designers from early times. Up to the present, little has been done toward automatically switching out an ailing part of the computer. The problem assumes a new order of importance when one considers the production of large functional aggregates by automatic means. For some proposed techniques, repair of a faulty element would be out of the question, and it would be essential to by-pass the element. It may be expected that when such aggregates become available, the "graceful degradation" property may be more easily realized since the large number of elements required will not be so difficult to provide.

In conclusion, forseeable improvements in hardware are, in the main, expected to affect reliability and economy, but improvements in speed and other performance capabilities will be made by advances in the logical organization of the computing system.

Walter Orvedahl
Inst. for Computer Research
University of Chicago
Chicago, Ill.

R61-97 Parallelism in Computer Organization Random Number Generation in the Fixed-Plus-Variable Computer System—M. Aoki, G. Estren, and T. Tang. (*Proc. Western Joint Computer Conf.*, Los Angeles, Calif., May 9–11, 1961; pp. 157–172.)

This paper considers in detail one use of a readily alterable special-purpose module (*i.e.*, a *variable* structure) in conjunction with a general-purpose computer (*i.e.*, a *fixed* structure). The claim is made—and well documented for the random number generation case considered—that the simultaneity made possible by such a fixed-plus-variable structure can appreciably reduce the solution time of certain problems.

As a preface to the actual design of a serial pseudo-random number generator, the authors first review in a concise manner the methods for producing and testing pseudo-random sequences. They also include some time statistics obtained by producing such numbers by program on an IBM 709 computer. A serial random-number generator design, using a multiplicative congruence method, is presented which, by using 5-Mc flip-flops, can complete the generation of a number in 8 μsec. A parallel generator is considered only briefly on the grounds that, due to the simultaneous nature of the generation, the

less complicated serial device would satisfy present and anticipated needs.

Three Monte Carlo applications are considered with primary emphasis on a transistor circuit design program which simulates a circuit by approximating the distribution of parameters and then selecting values of the parameters at random.

Several paragraphs are devoted to the fixed-plus-variable structures in general with the implications that similar advantages can be derived from considering such functions as $\ln x$ and e^x in the same manner. This is not obvious from the special case considered. The generation of pseudo-random numbers (and its antithesis a real time clock), are the only ready examples of functions which do not require, after initialization, any input parameters in order to produce an output. Thus, for example, the $\ln x$ procedure cannot start before a value for x has been received, and, in general (without the addition of considerable program complexity and program-interrupt devices), the general-purpose program cannot proceed until $\ln x$ has been obtained.

Bruce Arden
Dept. of Math.
University of Michigan
Ann Arbor, Mich.

R61-98 A Survey of Digital Methods for Radar Data Processing—F. H. Krantz and W. D. Murray. (*Proc. Eastern Joint Computer Conf.*, New York, N. Y., December 13–15, 1960; pp. 67–82.)

We quote the authors' summary:

This paper reviews the growing number of declassified techniques for automatic processing of radar data by digital means. Emphasis is placed upon signal time-sampling and quantization, integration methods, rejection of stationary targets, radar trigger manipulation, and treatment of radar beacon code data. These techniques are discussed individually and are also shown combined in a hypothetical radar data processor design.

The discussion in the paper is entirely qualitative and not very much detailed. It presupposes very little knowledge of radar on the part of the reader. Since there are no references at all, there is little help for a reader who is concerned with actual design of radar data processing equipment.

William L. Root
Lincoln Lab.
Mass. Inst. Tech.
Lexington, Mass.

E. Circuits and Components

R61-99 Digital Applications of Magnetic Devices—Albert J. Meyerhoff, Sr. Ed. (John Wiley & Sons, Inc., New York, N. Y.; 1960. 604 pp.+xix pp.)

This book fills an important gap in the previously available coverage of modern digital circuit techniques. The authors present material which was only available scattered throughout a large number of technical journals and convention records. The intended audience for the book is made up of digital circuit designers and users and system designers. The book provides information which will permit the system and circuit designer to improve his judgment in determining the application areas in which magnetic logic can be used to advantage. In addition, sufficient design detail is included to permit circuit designers unfamiliar with this field to design magnetic logic circuits.

The book is divided into seven major parts.

1) Fundamentals—fundamental magnetism, descriptive engineering parameters and equivalent circuits.
2) Parallel Magnetic Amplifiers—data processing circuits of the two core per bit type with parallel output.
3) Delay Parallel Magnetic Amplifiers—data processing circuits of the single core per bit type with parallel output.
4) Series Magnetic Amplifiers—data processing circuits using using series output and two cores per bit.
5) Memories—coincident current memories, linear selection memories, advanced memories.
6) Transistor-Magnetic Pulse Amplifiers—data processing circuits using magnetic logic and transistor amplification.
7) Other Techniques—carrier magnetic amplifiers and multiple aperture diodeless magnetic logic (MAD).

In each major section, the coverage is broken down into categories such as circuit design, logical design, system aspects, drivers, etc.

Unfortunately, the book suffers from unbalance in the coverage of the topics. For example, 113 pages are devoted to the two core per bit parallel output circuits of Section II and only 47 pages to the single core per bit circuits of Section III. The coverage in Section II provides sufficient design information but Section III is very inadequate in this respect. This is unfortunate because of the importance of the single core per bit circuits and the complexity of the circuit design. The treatment of the series output circuits and the transistor-magnetic circuits includes adequate design information. There is no discussion of the current steering circuits developed by Karnaugh. The description of carrier magnetic amplifier circuits and of MAD logic is general in nature. There is no description of other methods of performing diodeless magnetic logic or of other special-purpose multiple-aperture configurations.

The general organization of the book facilitates rapid access to particular design information. The index is good and each major section has been thoughtfully organized.

The book is an important contribution to the field.

Edward P. Stabler
Electronics Lab.
GE Co.
Syracuse, N. Y.

R61-100 Poly Aperture Cores Used in Non-Destructive Readout Counter—W. R. Johnston. (*Proc. 5th Annual Conf. on Non-Linear Magnetics and Magnetic Amplifiers*, Philadelphia, Pa., October 26–28, 1960.

In an interesting paper, the author discusses a new approach to to the design of blocking oscillator-type counters. The new feature in the circuits discussed is the addition of a multi-aperture memory element to the blocking oscillator. Using multi-aperture devices, nondestructive readout information is possible. First, the basic counter circuit is discussed, and, by verbal means, explanations of the manner in which the circuit operates are indicated. After careful reading, it was possible to see how the circuit behaves. It might be suggested that an analytical approach along existing blocking oscillator design procedures might be attempted. This would help to separate the storing procedures in the transfluxor used, and the transfer of information through the blocking oscillator action. As in any core circuit, a reset winding is required to advance the information stored in the multi-aperture device. Simple calculations would have shown that the blocking oscillator functions properly. The limitations of a single stage depend mainly on the selection of the magnetic material. This leads to the requirement of accurate control of the pulse width.

Cascading this special memory-type blocking oscillator allows the design of binary counters. An additional transfluxor and a diode, as well as wired-in logic, allow the design of decimal-coded counters with a minimum amount of components added. The speed of both counter types depends on the number of stages used, as well as on the propagation time and the pulse width. Since the delay time and the pulse width are controlled (as indicated above) by the selected material, the speed of a particular counter depends mainly on its size and the magnetic material used. An eight-stage counter will run at a repetition rate of 100 kc, and a counter of 22 stages at 40 kc, which means that a delay of about 4 μsec per stage has to be assumed. Different driving arrangements allow substantial reduction of this delay per stage. The memory-type blocking oscillator was used in a matrix scanning system.

The new feature of the counter is found in combining memory and switching action into one unit. In addition, the use of multi-aperture devices allows nondestructive readout. The claim of increased reliability is true insofar as only one transistor per stage is used. Nevertheless, the weakest link in the counter stage is still the semiconductor. Since, in missile applications, the mean free error time is short, the claim of improved reliability is adequate. A real improvement is reliability would be found in all-magnetic circuits. It is felt that Johnston's interesting paper is a step towards this ultimate goal.

F. F. Stucki
Missiles and Space Div.
Lockheed Aircraft Corp.
Palo Alto, Calif.

R61-101 Transient Analysis of Cryotron Networks by Computer Simulation—M. K. Haynes. (PROC. IRE, vol. 49, pp. 245–257; January, 1961.)

This paper briefly reviews the concepts of tensor methods as expounded by Kron and points out the important requirements for the solution of transient networks containing nonlinear elements. The lumped-constant *L-R* network representation of a cryotron is next given followed by a brief description of the program developed for the IBM 704. Necessary circuit description requirements for the program are presented.

As examples of the application of the tensor methods and the program, three examples of simple cryotron circuits analyzed on the 704 are presented. The first of these is a cross-latched cryotron flip-flop. The curves that are plotted from the computer output show clearly the proper values of the supply current for minimum switching time. Other design criteria can also be inferred from outputs obtained by rerunning the program for different values of some of the variables. The second example is a five-stage free-running ring circuit. The most noteworthy feature of the results of this analysis is the indication of a stable region for supply currents of greater than twice the threshold. This is not obvious from a knowledge of the circuit operation. The third example is a three-bit self-timing self-checking binary adder. This is by far the most complicated circuit analyzer. This reviewer understands from the author that some care was required in setting up this problem for the IBM 704 as the memory was just large enough to handle a problem of this magnitude. The results of the analysis as presented in the paper are sketchy but do indicate the power of the method.

In the opinion of this reviewer, the importance of this paper is that it points the way to what will become an absolute necessity as integrated circuits containing many hundreds of components are required. That is, the use of computers not only to analyze the circuits but actually to design them, and in the case of those fabricated by procedures related to vacuum evaporation, to prepare output tapes that can be used to prepare the necessary masks on automatic machines. It is virtually impossible to analyze a cryotron circuit as complicated as the three-bit adder example given in this paper without a computer. A next obvious step will be to extend a program such as the one described so that the computer modifies the orignal input data and then prepares the masks tapes.

The author of the paper has indicated to this reviewer that the 704 program described can be obtained by those desiring it. My own reservations with regard to the actual program are that no allowances are made for transition times of the gates, thermal effects, and several other effects that are sometimes of extreme importance in actual cryotron circuits. These effects are, of course, extremely difficult to describe mathematically and their omission is merely an indication of the large amount of research that remains to be done on cryotrons, rather than an oversight by the author.

LESLIE L. BURNS, JR.
RCA Labs.
Princeton, N. J.

R61-102 Cryotron Storage and Logical Circuits—M. K. Haynes. (*Solid State Electronics*, vol. 1, pp. 399–408; September, 1960.)

In this well-written article, the author describes the use of persistent supercurrent storage elements in a number of potential memory arrangements. He also describes the use of this type of element in arithmetic and logical circuits.

Specifically, he describes the use of persistent supercurrent elements in memory arrangements in which reading and writing can be performed simultaneously, in which transfer of information from one word to another in a memory can be achieved internally, and in which multiple access to the memory can be achieved. He also describes how the elements may be used for shift registers, counters, adders and a number of logical elements.

Although the article describes clearly many of the potential applications of these types of cryotron devices, it does not indicate the present feasibility of some of the implied fabrication techniques or the uniformity problems. Unfortunately, the author has not included any references to other work in the field.

ARTHUR V. POHM
Elec. Engrg. Dept.
Iowa State University
Ames, Iowa

R61-103 A Computer Subsystem Using Kilomegacycle Subharmonic Oscillators—I. Abeyta, F. Borgini, and D. R. Crosby. (PROC. IRE, vol. 49, pp. 128–135; January, 1961.)

The phase-locked oscillator (PLO) has received considerable attention as a potential logic element in digital computers operating at extremely high speeds. Several papers (referenced in the review paper) have dealt with the theoretical aspects of nonlinear subharmonic oscillators. This paper presents the results of experiments on a limited number of varactor PLO's operating with a pump frequency of 3.7 kMc and a signal frequency of 1.85 kMc. The pump is modulated with a 30-Mc squarewave. Since the conventional three-phase clocking technique is utilized in order to make the device operate unilaterally, the equivalent logic rate is 11 nsec.

The author describes the physical construction of the PLO microwave structure and then proceeds to describe a simplified lumped-parameter equivalent circuit which neglects the nonlinear varactor capacitance. It was stated that the PLO inclosure was designed for a characteristic impedance of 91 ohms to achieve a small size. However, it is pointed out that an input impedance mismatch to the PLO results in measured standing wave ratio of 8 to 12, and the calculated standing wave ratio from the lumped-parameter equivalent circuit is stated as being between 9 and 10. Therefore, a matching network is required with consequent loss in pump efficiency. Since this mismatch contributes so much to the limitations of the use of PLO's, the question arises as to whether a more optimum mounting structure could have been designed to result in a better SWR.

The paper points out the relatively complex problems that occur in a microwave PLO system—in particular, the need for matching input and output impedances to reduce or eliminate spurious oscillation and the resulting effect of inefficient utilization of pump power. Reference is made to the generation of spurious frequencies at several points; however, no specific data or relationship of the spurious frequencies to the pump or signal frequency are given.

In general, the paper presents an informative and interesting discussion of the difficulties associated with microwave PLO logic and indicates that major engineering problems must be solved before PLO's can be utilized in kilomegacycle computers.

GEORGE WOLFE, JR.
Ground Systems Group
Hughes Aircraft Co.
Fullerton, Calif.

R61-104 Electronics and Nucleonics Dictionary—N. M. Cooke and J. Markus. (McGraw-Hill Book Co., Inc., New York, N. Y.; 1960. 543 pages.)

This work is actually a second edition of the "Electronics Dictionary" published fifteen years ago. The broadened scope of the title represents the increased coverage of the book, which now includes some 12,000 definitions. The authors are to be congratulated on many counts—accuracy, coverage, simplicity, and style of presentation are all excellent.

A dictionary of this sort is intended for many classes of users. Editors, authors, and secretaries will find it a valuable source on matters of spelling, hyphenation, capitalization, and usage. Engineers and scientists outside the fields of electronics and nucleonics will find the definitions easy to understand, and those in one specialized area of electronics will find the definitions in other areas valuable. Dictionaries are probably not intended, however, for the specialist who wants a complete and precise definition of terms within his own area of specialization. Thus, the computer engineer will not turn to this book for definitions of computer terms, since he will probably find them not sufficiently detailed. For example, we find that a shift register is "a computer circuit that converts a sequence of input signals into a parallel binary number or vice versa." Such a definition might be highly informative to an audio engineer but conveys little new to a computer engineer. However, for acquainting one quickly and effectively with terms outside his own area, this book should prove very useful.

NORMAN R. SCOTT
Elec. Engrg. Dept.
University of Michigan
Ann Arbor, Mich.

F. Memories and Access Circuits

R61-105 Magnetic Film Memory Design—J. I. Raffel, T. S. Crowther, A. H. Anderson, and T. O. Herndon. (Proc. IRE, vol. 49, pp. 155–164; January, 1961.)

An excellent review of the basic properties of ferromagnetic films with uniaxial anisotropy is presented in this paper. The factors which contribute to control of the uniaxial anisotropy of permalloy films prepared both by vacuum deposition and by electroplating are treated in considerable detail. Especially important is the treatment of the problem of "dispersion," or the variation of the easy direction of magnetization within a given film element. This most important consideration has been overlooked in many of the earlier papers on thin ferromagnetic films.

The design and performance of a word-organized memory employing nonmagnetostrictive permalloy storage elements is presented. It is implied that, with sufficient control of magnetic properties of the film elements, this design will permit the nondestructive sensing of the state of the film storage elements by subjecting their magnetization to reversible rotation. Other more immediately available nondestructive readout memories can be constructed employing storage cells, each comprised of two film elements of different coercivity.[1]

The specifications of the TX-2 Thin Film Memory consisting of 32 ten-bit words with an operating cycle time of 0.8 μsec are given. The authors state that this device has been bench-tested at a 0.4-μsec cycle time. There follows a complete and stimulating discussion of the problems of extending the storage capacity and decreasing the cycle time of memories utilizing this design.

Sidney M. Rubens
Remington Rand Univac Div.
Sperry Rand Corp.
St. Paul, Minn.

[1] L. J. Oakland and T. D. Rossing, "Coincident-Current nondestructive readout from thin magnetic films," *J. Appl. Phys.*, vol. 30, suppl., pp. 54S–55S; April, 1959.

R61-106 The "Persistor"—A Superconducting Memory Element—E. C. Crittenden, Jr., J. N. Cooper, and F. W. Schmidlin. (Proc. IRE, vol. 48, pp. 1233–1246; July, 1960.)

The Persistor is a superconducting memory element in which current pulses of one polarity on the drive lines are utilized for writing "1's" and current pulses of the opposite polarity on the same drive lines are utilized for writing "0's" and reading information out of memory.

As explained in this paper, the operation of the Persistor relies upon the current magnitudes at which certain things occur; *e.g.*, the current magnitude at which superconductivity first disappears (resistance first appears) is referred to as the threshold current (I_t), and the minimum pulse-current magnitude at which the entire Persistor switches simultaneously is referred to as the simultaneous switch current (I_s). A third current magnitude is referred to as the dc critical current and is shown in Figs. 11–13, and 15–17, but this current is not defined; however, it appears to be the minimum dc current at which full resistance appears (in any manner) in the film being tested. More emphasis is placed on the dc critical current than can be justified since the dc critical current does not appear in the operation of the Persistor, the Persistor operation being based entirely upon the threshold current and the simultaneous switch currents [(1)–(6), page 1242]. The threshold currents which are shown were determined through the use of a pulse technique with a sensitivity of 50 μv/cm; however, these same currents can be determined to a greater accuracy through the use of dc techniques, and sensitivities as high as 0.1 μv/cm are readily obtainable.

A discussion of multidimensional memory arrays utilizing Persistors is presented on pages 1241–1244, but it is not immediately clear to the reader which part of the discussion pertains to memories in general and which part is restricted to memories in which the word length cannot be greater than one bit. As an example, consider the statement (page 1241), "If N is the number of Persistors in the cryostat, only $3\sqrt[3]{N}+1$ leads are needed for three dimensional operation." This statement is true if the leads are two-conductor cables, and if the word length is only one bit. For a three-dimensional memory containing N persistors with M words and P bits per word ($N=MP$), the number of leads (2-conductor cables) into the cryostat is $2\sqrt[3]{M}+2P$ where $\sqrt[3]{M}$ leads are required for X selection, $\sqrt[3]{M}$ leads are required for Y selection, P leads are required for Z selection, and P leads are required for the sense output. For the million Persistor memory mentioned on page 1242, $3\sqrt[3]{N}+1=301$ leads, but if $P=\sqrt[3]{M}=100$, then $2\sqrt[3]{M}+2P=400$ leads.

Approximately one page of detailed explanation of the theoretical Persistor operation is presented on pages 1234 and 1235. This explanation is clear and is easy to follow. An explanation of the waveforms in Fig. 25 and the experimental Persistor from which these waveforms were obtained occupies a small portion of page 1242 and is not in sufficient detail to be understood by the reader. This is unfortunate because this information is not presented elsewhere, and the reader is deprived of knowing and understanding some of the intricacies of Persistor operation and testing.

The statement is made on page 1245 that neither indium nor tin can be satisfactorily deposited on glass substrates at room temperature, but what is not satisfactory about these films is not stated. The statement, "The problem seems to be the nucleation of an insufficient surface density of initial metal crystals on the surface," indicates that the films are discontinuous. Thin-film cryotrons utilizing tin and indium films, deposited onto room temperature substrates, with thicknesses of a few tenths of a micron have been satisfactorily fabricated and tested.

Many good illustrations are shown in this paper, but Figs. 4, 8, and 22 did not meet the standard set by the others. In Fig. 4 there is incorrect curvature of the portion of the I_R waveform which occurs as a result of the trailing edge of the pulses of the I waveform. Identifying callouts would have made Fig. 8 much easier to understand. In Fig. 22 the sense line misses one of the memory cells.

Two typographical errors were noted on page 1243. In the text $I_S=R=0.30$ volt should be $I_SR=0.30$ volt. In the caption of Fig. 27 "text pattern" should be "test pattern."

This paper, which describes in great detail the state of the art of superconductive film development and the development of the Persistor, makes an excellent addition to the literature on superconductive devices.

James P. Beesley
Federal Systems Div.
IBM Command Control Ctr.
Kingston, N. Y.

R61-107 The Development of a Multiaperture Reluctance Switch—A. W. Vinal. (*Proc. Western Joint Computer Conf.*, Los Angeles, Calif., May 9–11, 1961; pp. 443–475.)

This paper consists mainly of 1) a review of some basic switching properties of two-aperture magnetic cores, 2) a description of a "subtle switching phenomenon" termed "inner wall reflex switching," and 3) a brief description of a three-dimensional memory array employing the keyhole Multiple Apertured Reluctance Switch (MARS). It appears that there is nothing significantly new in this paper, except possibly the detailed shape of the MARS and some memory wiring techniques (described in less than one page out of ten in the text).

In describing core properties, the author introduces some uncommon terms (*e.g.*, vortex source, coherent energy, geomechanics) that do not seem to be of any aid to clarification or understanding.

Certain well-known facts about core properties (*e.g.*, the fact that threshold drive for switching flux around the "control" aperture is proportional to its perimeter and independent of its distance from the "read" aperture) are presented as significant experimental conclusions (almost as if the author were surprised by such results).

"Inner wall reflex switching" is the same switching phenomenon that occurs when any transfluxor-like core is unblocked by excitation of a drive winding on the outer leg adjacent to a minor (or "read") aperture.[1,2] The author presents experimental data to support his explanation (which includes some elegant flux-path diagrams) of this phenomenon. It is to be seriously questioned whether these data actually provide any significant verification of the details of the explanation given.

D. R. Bennion
Stanford Research Inst.
Menlo Park, Calif.

[1] S. A. Abbas and D. L. Critchlow, "Calculation of Flux Patterns in Ferrite Multi-Path Structures," 1958 IRE National Convention Record, pt. 4, pp. 263–267.

[2] S. A. Abbas, "Methods of Analysis of Flux Patterns in Ferrite Multipath Cores," Dept. of Elec. Engrg., Carnegie Inst. of Tech., Pittsburgh, Pa., Final Rept., IBM Purchase Order L-7039; June, 1958.

R61-108 Magnetic Tape Instrumentation—Gomer L. Davies. (McGraw-Hill Book Co., Inc., New York, N. Y.; 1961. 241 pp. and 9 index pp. +9 pp. appendix; illustrated. $8.50.)

This book is a collection of a large amount of information pertaining to the use of magnetic-recording techniques. It is well organized and easy to read if one has a basic understanding of the principles involved and some experience with the use of magnetic-recording equipment.

The emphasis is clearly on the use of tape for the recording of analog data in the form of a frequency-modulated signal, and all aspects of this form of utilzation of magnetic storage are well covered.

The book does not, as no doubt the author intended, cover very completely the use of magnetic recording in other fields, although there is given at least an introduction to the use of magnetic drums, tape and disks as digital recorders.

Descriptions of circuit configurations useful in connection with magnetic recording are limited chiefly to block diagrams and skeletonized schematics. No attempt is made to describe practical circuitry. Coverage of the mechanical aspects of tape transport design is limited to descriptive material.

References to publications giving greater detail are included at the end of each chapter and in themselves provide a source of information which would require considerable effort to uncover.

In all, the book is well done and worth reading if one is interested in the use of magnetic tape as an adjunct to analog instrumentation, but is not of high value if the field of interest is digital application, circuits or circuit techniques.

F. G. Buhrendorf
Bell Telephone Labs., Inc.
Murray Hill, N. J.

G. Programming and Numerical Methods

R61-109 Steps Toward Artificial Intelligence—Marvin Minsky. (Proc. IRE, vol. 49, pp. 8–30; January, 1961.)

In the paper under review, the author presents an argument to defend his thesis that

A computer can do, in a sense, only what it is told to do. But even when we do not know exactly how to solve a certain problem, we may program a machine to "Search" through some large space of solution attempts. Unfortunately, when we write a straightforward program for such a search (sic), we usually find the resulting process to be enormously inefficient. With "Pattern-Recognition" techniques, efficiency can be greatly improved by restricting the machine to use its methods only on the kind of attempts for which they are appropriate. And with "Learning", efficiency is further improved by directing Search in accord with earlier experiences. By actually analyzing the situation, using what we call "Planning" methods, the machine may obtain a really fundamental improvement by replacing the originally given Search by a much smaller, more appropriate exploration. Finally, in the section on "Induction", we consider some rather more global concepts of how one might obtain intelligent machine behavior.

The reviewer agrees with the spirit but not the substance of the author's remark on Search, (footnote 3 on page 9) "(However, I am not convinced of the usefulness of his [Ashby's] notion of 'ultrastability,' . . .)." Ultrastability is a *formula* expressing ancient and well-known results; it is also a teaching device and therein lies its usefulness. However it is to be agreed that the idea of ultrastability can be taken too seriously.

Concerning "property lists" the reviewer is puzzled by the author's remark on the bottom of page 12, "We define a *property* to be a two-valued function . . . ." However, this is like saying "the shortest distance between two points is a straight line"—but how can a distance be a straight line even in Euclidean geometry?

On page 23 in Fig. 11, he considers Shannon's 2-person game played with a 2-terminal resistive network in which one player aims at having a short circuit between the two terminals and the other player tries to get an open circuit between the two terminals. A player's move consists of either opening or shorting one of the resistors in the 2-terminal network. Motivated by this idea, the reviewer has played a similar game with a few 2-terminal RLC networks in which a move consists in either opening or shorting one of the components or making an alteration of complex electrical frequency.

The paper terminates with 95 references to the literature; all of these are used in developing the text, rather than just being mentioned. However, the reviewer believes that a reference to the recent book by Braines, *et al.*, should have been included.[1] The paper under review is sufficiently broad to warrant the careful attention of the engineering community.

Errata: In the right-hand column of page 11, omit lines 19 through 22, inclusive, counting from the bottom of the page; they are a redundant proper subset of the sentence that follows. In the left-hand column of page 12 on the last line read "distinct" for "distinction" and N for n. In the right-hand column of page 12 on the second line read N for n. In the left-hand column of page 27 on the 13th line from the bottom read "Gödel" for "Godel."

A. A. Mullin
University of Illinois
Urbana, Ill.

[1] S. N. Braines, A. V. Napalkov, and V. B. Svechinskii, "Problems of Neurocybernetics" (transl. from Russian by D. P. Barrett), A. M. Andrew, Ed., National Lending Library for Science and Technology, London, Eng.; October, 1960.

R61-110 A Basis for a Mathematical Theory of Computation—John McCarthy. (*Proc. Western Joint Computer Conf.*, Los Angeles, Calif.; May 9–11, 1961; pp. 225–238.)

This is one of the most skilfully written technical papers the reviewer has ever read. Its purpose is to establish a basis for the science of how computing machines and general automata carry out intellectual processes.

McCarthy believes that "the limitations on what we have been able to make computers do so far . . . come far more from our weakness as programmers than from the intrinsic limitations of machines." His hope is that a mathematical theory of computation might enable us to overcome many of these limitations.

He mentions the following goals for such a theory: the development of a universal programming language; the definition of a theory of equivalence among computation processes; the representation of algorithms by symbolic expressions in such a way that significant changes in the behavior represented by the algorithms are represented by simple changes in the symbolic expressions (to treat learning and growing phenomena); the representation of computers as well as computations in a formalism that permits rigorous treatment of the relation between them; and the construction of a quantitative theory of computation (for measuring the size and reliability of computations). The paper contributes a great deal to several of these goals, and is, apparently, at least compatible with all of them.

McCarthy's main contributions are: 1) a reformulation of recursive function theory using a conditional form such that its domain is completely generalized and its recursive definitions are much easier to handle than before; and 2) the development of a method called recursion induction for proving equivalences between algorithms defined under the recursive formalism of 1). The first contribution meets the obvious requirement that any general theory of computation must fit into the framework of recursive function theory, yet be free of the awful encumberances of that theory. It also avoids a weakness of ALGOL and most other modern programming languages by having built into it an ability for describing different kinds of data spaces. The second contribution is a crucial followup of the first: it shows that reasonably powerful proof techniques can be developed within the proposed formalism.

The paper is divided into two main parts. The first contains sections on the following:

1) How new functions and forms can be composed from base functions defined over arbitrary base spaces.
2) The recursive function theoretic notion of partiality, whereby the values of functions may be explicitly or implicitly undefined over parts of their domains. This concept is probably crucial for the manipulation of heuristic-type programs.
3) Predicates and propositional forms.
4) The conditional form $(p_1 \rightarrow e_1,\ p_2 \rightarrow e_2, \cdots, p_n \rightarrow e_n)$. This modified ALGOL expression, when defined, is equal to the e corresponding to the leftmost p (proposition) that is true. The great power of the form is that it can control the flow of calculations directly instead of forcing them to be controlled by cumbersome integer calculations as in recursive function

theory. It also more closely aligns computability theory with program flow-chart theory and logical system design.

5) Definitions of functions by recursion using conditional forms.
6) A demonstration that computable functions are much simpler to handle in the new formalism, based on the successor function, predicate equality, and the conditional form, than in the old formalism of recursive function theory.
7) Noncomputable functions. By adjoining quantifiers to the previous formalism, McCarthy admits the full run of unsolvability results. But his emphasis is on functions not *a priori* computable but which can be shown equivalent to computable ones. His hope is "that the mathematics of computation may have as one of its major aspects rules for transforming functions from noncomputable to computable form."
8) Ambiguous functions. Functions like "less than x," "divisible by y," etc., which are not completely specified, seem promising devices for proving convergence and other facts about recursion formulas where certain information is irrelevant.
9) A set-theoretic approach to the extended definition of new data spaces and functions in terms of given data spaces and functions.

The second part of the paper contains sections on:

1) The formal properties of computable functions. Here the mechanics of proving partial and completely defined functions equivalent by reverting their expressions to canonical form is discussed.
2) Recursion induction. This is a method for proving the equivalence of a pair of functions g and h by constructing a function f satisfied by both g and h over their domains.
3) The relations between McCarthy's formalisms and those current in programming and recursive function theory.

McCarthy warns us that his basis still has some syntactic deficiencies which render its proof schemes relatively incomplete. But this is to be expected at such an early stage. The real concern is whether or not modern mathematical logic is adequate to the task of overcoming present intuitive limitations. In certain restricted areas, of course, it is. But recall for example that we still know nothing basic about the logic of relations: simple things like *A thinks B is C*, *A gives B for C*, etc. Yet humans think in these terms with marvelous facility, so why shouldn't we program in them with equal facility, at least occasionally. I think the answer is that we do. Witness, for example, the industrial dynamics programs at M.I.T.'s School of Industrial Management.

McCarthy's paper is strictly top rank, but the reviewer questions just how general our purportedly generalized theories of computation can get at this state.

W. L. Kilmer
Elec. Engrg. Dept.
Montana State College
Bozeman, Mont.

R61-111 Annual Review in Automatic Programming, Vol. I—Richard Goodman, Ed. (Pergamon Press, Inc., New York, N. Y.; 1960. 229 pp.+xi pp.+70 Appendix pp. $10.00.)

An important outcome of the Conference on Automatic Programming of Digital Computers held at the Brighton Technical College on April 1–3, 1959, was the decision to establish an Automatic Programming Information Center at Brighton with the aims of establishing a permanent collection of material on automatic programming, publishing the "Annual Review in Automatic Programming," and organizing conferences on particular aspects of the subject. Since the conference itself proposed to attempt a first review of the work being done, to afford a chance for an exchange of views on problems and difficulties, and to provide an authoritative and readily available record of the then current position, the well-prepared, intelligently-edited and beautifully-printed "Annual Review" provides one of the answers. Included are eighteen papers, five of which are either summaries of material presented at the conference or general statements concerning some aspect of the fields of automatic programming and automatic coding. The remaining thirteen papers may be considered as technical reports on specifications, on operational experience or on problem-solving techniques of automatic coding systems. Finally, the appendixes include A. M. Turing's two pioneer papers, a preliminary report on ALGOL, and a short bibliography.

Due to the rapidity of the developments in the field of automatic coding, many of the papers have become out-dated. However, it is most useful to have a record of "early" opinions and results both for historical reference and for rechecking the reasons for and against techniques and directions which were or were not implemented. Further, it is clear that in England and in Europe, developments in this field, particularly in the area of business data processing, have lagged behind developments in the United States, and that—unfortunately—with no periodical or other publication devoted to programming information, communication of achievement in this field has been less than satisfactory.

In the "Opening Address," Dr. A. D. Booth traced the historical development of automatic coding and programming, examined the circumstances under which the techniques were useful or required, and restated advantages and disadvantages. Time has confirmed the advantages he listed, they are now "common sense." Of the disadvantages he listed, many have been overcome in the intervening period, some were trivial, others were indicative of the varying degrees of suitability of the available techniques as applied to different types of problems. The second paper by A. E. Glennie on "Future Trends in Automatic Programming" correctly predicted the use of "declarations," and "standard numerical procedures" as two schemes leading to the development of programming languages. However, it missed the fact that future improvements will need many other schemes and techniques which might well fall outside of the two cases defined and illustrated by the author.

K. A. Redish, in his paper, "Some Problems of a Universal Autocode," pleads for common programming languages particularly in the field of business data processing. Within less than two years, his problem is well on the way toward being solved by the COBOL effort in the United States. Progress toward a common scientific language is evident from the number of ALGOL translators now extant. The speed of these developments has out-dated the paper before it was published. However, it will serve to recall the basic aims, purposes, and reasons for "natural" programming languages.

The paper by Dr. Stanley Gill on "The Philosophy of Programming" is a rather strange mixture. The first half of the paper offers a brilliant description of the processes and history of programming efforts. The second half of the paper is clumsy. Words and terms are used without careful definitions and in incorrect context. The paper clearly shows the great need for a standard set of definitions of such sets of terms as systems analyst, programmer, coder, or assembly, compiler, generator, translator. Dr. Gill's touches of humor make any paper he writes well worth reading.

Again, in the case of Gordon Cushing's paper on "Automatic Programming and Business Applications," developments and projects in the United States were even then ahead of those reported. However, his discussion of the nature of the programming task will be worth rereading not only next month, but also five years from now as a yardstick of accomplishment.

The thirteen technical papers vary considerably in their comprehensiveness and sophistication. They will provide a "fix" for historical reference and it is well to have descriptions such as these available to all as well as to those in the inner circle who exchange manuals. Of particular interest is the paper by Pearcey, Higgins and Woodward on "The Mark 5 System of Automatic Coding for the TREAC." It should encourage those concerned with designing techniques for very small computers. The description of the programs for the Pegasus computer is an excellent example of the presentation of a complex system. Together with Payne's and Ronaldson's descriptions of experience using the system, a very clear picture is given of an operating system. Ronaldson's paper is a model of what a really excellent technical report can be.

J. P. Cleave's report on the application of formula translation to the "Automatic Coding of Ordinary Differential Equations" covers a delightful and clever invention by which an inefficient compiler can be circumvented. The three-address coding is a useful tool, and knowledge of it should be included in the tool kit of all automatic programmers. Again, R. A. Booker's description of the "Mercury Autocode: Principles of the Program Library" includes the treatment of some dynamic devices and intelligent variables which undoubtedly will be useful in the development of automatic programming techniques as contrasted to those of automatic coding.

Of the two papers on auto-coding for the DEUCE, the first by C. Robinson is a description of a collection of computing and utility routines rather than of a system. The second by S. J. M. Denison de-

scribes four minor systems and some additional routines. It would be a happier circumstance if these routines had each formed a part of an integrated system together with an allied system of service routines.

"The STANTEC-ZEBRA Simple Code and Its Interpretation," by R. J. Ord-Smith, is truly simple enough for the nonexpert to use and is flexible enough to execute the bulk of the elementary mathematical operations. The paper should be read as a reminder that simple codes are required and can be developed. The report by K. V. Hanford on "The Share Operating System for the IBM 709" is so simple and clear that it resembles a bland diet. It is too simplified to discuss any problems, difficulties or new techniques encountered. In contrast, Alan Taylor makes use of a sharp pen in describing the problems encountered with respect to the acceptance of automatic programming, its pros and cons, and the aptitudes of the users, in his paper on the FLOW-MATIC and MATH-MATIC programming systems. While both systems have undergone considerable evolutionary development since the paper was written, it provides a clear introduction to them. "Tide: A Commercial Compiler for the IBM 650" is not a true automatic programming system. Rather it is a quickly and expediently built system to generate limited types of programs. As Humby indicates, it was assembled on the skeletons of selected previously-tested programs and as such is both easy to use and easy to amend. J. E. Meggetts' paper on "Auto-Programming for Numerically Controlled Machine Tools" is a clear and precise description of a somewhat familiar operation.

Again considering the book as a whole: the editor has made the book a pleasure to read with excellent typography. The inclusion of the two pioneer papers on "Computable Numbers" by the late Dr. A. M. Turing would, by themselves, make the book essential to any good programming library. Adding the ALGOL report and the short bibliography was a most thoughtful idea. All personnel working in the field of automatic programming and automatic coding should have the opportunity of reading Turing's papers. ALGOL is a language which should be known to all scientific programmers.

A few other general comments must be made:

1) Almost all of the automatic coding systems reported (as well as the computers for which they are designed) are oriented toward mathematical rather than industrial data processing. As the computers enter the industrial, accounting and commercial fields in Europe, it will be interesting to see what new contribution will be made.

2) It is confusing to find a particular term used to refer to different things, and different terms used for the same thing. The need for a common glossary is clearly evident. Perhaps the next volume will supply it.

3) The evident lack of communications between groups, companies and hemispheres in the field directs attention to the very great need for reviews such as this book and also for textbooks and other educational material covering the field of automatic coding.

4) Perhaps the next conference will be able to pay greater attention to the organic structure and dynamic functional relationships among the components to be integrated as unique automatic coding systems. Badly needed are the algorithms for building the best "components," the methods of integrating the components into a unique entity, and the techniques for optimizing latency, storage allocation, etc., and for balancing the structure of the whole system.

5) Also needed are criteria, yardsticks and meters to check the logical sequences, the redundancy, and the efficiency of the object programs.

Thus, the book under review supplies historical background, presents accomplishment, and points to areas requiring future development. It should be read by all who are concerned with research, design, implementation and application of automatic coding and automatic programming techniques.

Grace Murray Hopper
Remington Rand UNIVAC Div.
Sperry Rand Corp.
Blue Bell, Pa.

R61-112 Self-Organizing Systems—A Review and Commentary—J. K. Hawkins. (Proc. IRE, vol. 49, pp. 31–48; January, 1961.)

Systems called "self-organizing" are currently enjoying a vogue in inverse proportion to the precision of the term. In this article, an active worker in the field attempts to carve out a safe and sane subset of such systems by limiting his point of view to that of the digital-computer engineer, and by restricting his scope to that of recognition (fixed-parameter) and learning (adjustable-parameter) systems realizable as networks.

The general performance asked of such systems is to realize an arbitrary mapping from an input set to an output set. A recognition net can realize only one such mapping; a learning net is expected to alter its mapping toward an ideal, usually on the basis of an error signal considered to be separate from the rest of the input set and referred to the ideal mapping.

The art of recognition-network synthesis consists in the selection of input transformations that can be mapped more economically on the output than can the input set itself. The cognate art of learning nets involves also the selection of distributions of error information that will be most effective in altering the mappings convergently to the ideal.

In a historical summary, including a section on character-recognition devices (the utilitarian avatars of these nets), Hawkins describes in detail several of the more salient systems built or proposed. In a section on the problems of the field, he furnishes a formalization of his own and discusses several systems within its framework. After noting repeatedly the limitations of systems employing only linear operations on the input variables, he identifies the central problem of the field as the selection of appropriate nonlinear functions and their economical realization. He describes two techniques for generating such functions, both uneconomical.

In a brief section on sequential networks, the author speculates whether such networks might get us beyond present barriers but makes no commitment. He mentions "evolutionary" systems, in which the learning problem is to be solved by altering both the number and the connectivity of elements, *i.e.*, the number of states of the system, but he wisely reserves judgment on whether this is a difference that makes a difference.

The biological provenance of the problem and many of its solution techniques is noted by scattered acknowledgments and analogies throughout, and in a description of the abstract "neuron" employed explicitly by many workers. His treatment, nevertheless, indicates that the umbilical cord is withered, to the extent that when he speaks of "neurons," "conditioned reflexes" and "learning" he describes objects and behaviors that biological systems do not show. It would be a pity if the cord is cut too soon, before we realize that biological systems often do not show the behavior of which they are supposed to be the "existence proofs."

The further study of recognition and learning systems might well be more intimately informed by the strictures of Shannon's Tenth Theorem, or Ashby's cognate Law of Requisite Variety, both of which say, in effect, that any system which selects to a degree better than chance does so on the basis of information received. This suggests that much of the generalization and transfer we admire in biological systems does not indicate the presence of a mechanism for induction, but rather an evolutionarily adequate balance between choice and chance. It may well be that we are admiring as choice that which is chance.

George W. Zopf, Jr.
University of Illinois
Urbana, Ill.

H. Reliability

R61-113 Redundant Logic Circuitry—J. G. Tryon. (U. S. Patent No. 2,942,193.

Tryon presents a design procedure for improving the reliability of logical circuits by the redundant use of components. The components used are of the electronic type, such as AND and OR gates. The circuit is so designed that no malfunction occurs despite failure of any single component or failure of most pairs of components. Hence, if the probability of a component error is p, then the probability of a circuit error would be proportional to p^2 (provided that circuit components fail independently).

To introduce the redundancy, each gate in the nonredundant circuit is replaced by four gates, each of the four has twice as many inputs as the gate it replaces. To avoid error when a single gate operates incorrectly, a single bit is represented by the state of a bundle of four wires. The connections required between successive levels of logic are simple; the outputs of each of a set of four gates are connected to two different logic elements in the next level. The specific assignment of connections is dependent on the nature of the levels to be connected. For example, the pattern of connection from OR gate to AND gate differs from that of OR gate to OR gate.

It seems appropriate to compare Tyron's design technique with the "two out of three" majority logic technique. In the majority logic technique, each bit is represented by the state of the majority of three wires. Each gate in the nonredundant circuit is triplicated and the three-wire output is fed simultaneously to three majority gates used as "restoring organs" in the manner of von Neumann.[1] As in Tryon's design, the redundant circuit will function correctly if any single component fails and will tolerate the failure of most pairs of components. Hence, the majority logic and Tryon's technique will yield roughly the same reliability. The more recent technique will require about eight times as many components as the nonredundant circuit, since the number of gates will be quadrupled, and each gate will require twice as many inputs as will the nonredundant case. The majority logic will require approximately six times as many components if single-element "majority gates" (such as magnetic cores) are used. Hence, there seems to be no significant difference in performance between majority logic and Tryon's technique.

Whether redundancy should be used in a given design will depend, of course, on the specific application. No useful general statements can be made regarding this.

Edward Arthurs
Mass. Inst. Tech.
Cambridge, Mass.

[1] J. von Neumann, "Probabilistic logics," in "Automata Studies," Princeton University Press, Princeton, N. J.; 1956.

I. Analog Systems

R61-114 Analogue Computation—Stanley Fifer. (McGraw-Hill Book Company, Inc., New York, N. Y., 1961. 1293 pp., 4 vols. $39.50.)

"Analogue Computation," by Fifer, is not to be confused with other recent publications with similar titles, *i.e.*, "Analog Computation," by A. S. Jackson (R61-34) published 1960, "Analog Computation in Engineering Design," by A. E. Rogers and T. W. Connolly, (R61-34) published 1960, or "Analog Simulation," by W. J. Karplus and W. W. Soroka, published 1959, all by McGraw-Hill Book Company, Inc., New York, N. Y.

Fifer's contribution is a four-volume set which gives a truly comprehensive coverage of the subject. In addition to containing a compilation of all known analog computing techniques, it also contains an excellent bibliography. The first two volumes cover analog computing components and the fundamental applications of these devices. The second two volumes describe advanced techniques in the art of analog computation. Fifer's work is not only complete, but, in addition, has that rare quality of being readable. He is rigorous where rigor is needed and yet manages to make his text flow smoothly. This reviewer especially likes the approach used in Chapter 4, where the author posed several straightforward physical problems, derived their mathematical equations, solved these equations by classical techniques, and then demonstrated the ease with which these problems could be programmed and solved on an electronic analog computer. I know of no other device which demonstrates the value of a computer so succinctly.

The only shortcomings in Fifer's work are due to the inevitable time lag between the writing of the manuscript and the distribution of the book by the publisher. This has led to the exclusion of techniques which are known by the various names of DYSTAC (Dynamic Storage Analog Computer), IDA (Iterative Differential Analyzer), and Point Storage.

The author has devoted himself exclusively to Analog Computers and makes no mention of Hybrid Computers, Incremental Computers and combined analog-digital techniques.

If you have only one work on analog computation in your library, this should be it!

J. E. Sherman
Lockheed Missiles and Space Co.
Sunnyvale, Calif.

J. Codes

R61-115 Two Dimensional Parity Checking—Peter Calingaert. (*J. Assoc. Computing Machinery*, vol. 8, pp. 186–200; April, 1961.)

This paper describes some techniques for multiple-burst error correction applicable to computers and data transmission. These techniques apply where data bits can be organized into two-dimensional arrays—either physically, in parallel tracks on tape, parallel data channels, or arrays of memory elements, or, mathematically, by means of redistribution of data bits (*i.e.*, by programming). The technique uses simple burst error-correcting and error-detecting codes of Hamming, Abramson, Melas and Meggit to check the data by rows and by columns. These codes can easily be implemented by feedback shift registers or by programming. Where the computer logic is available for error-correcting, Bose and Ray-Chaudhuri codes can be used. The technique assumes that a buffer, such as the computer memory, or some accessible storage at a data terminal, is available for the array.

The power of this technique is that the minimum distance of the effective two-dimensional code is the product of the minimum distance of the codes used for row and column checking, respectively. Thus, if a Hamming SEC code plus a parity check is used on the columns ($d_c=4$) and on the rows ($d_r=4$), the effective minimum distance is $D=d_r \cdot d_c=16$. Then $(D-1)=15$ errors could be detected, or the largest integer contained in $[(D-1)/2]$ or 7 errors could be corrected. When D is even, $[(D-1)/2]$ errors can be corrected plus $D/2$ errors can be detected.

Procedures are given for parallel and serial decoding. The parallel decoding requires additional logic to complement positions in the intersections of row and column corrections having the same number of errors. Serial correction of row and then columns allows temporary erroneous corrections which are corrected by recomputing the column parity bits after the row correction. This avoids extra logic but requires additional time.

The system is applicable to groups of error bursts, defined as an areal burst of dimension k_r by k_c, or a $(k_r \times k_c)$ burst lying within the rectangle k_r by k_c. Using the Abramson SEC-DAEC code in both dimensions, it is demonstrated that certain configurations of three $(k_r \times k_c)$ bursts can be corrected. The correction of this configuration of multiple bursts appears useful for defects affecting spots or sections in computer memory. The usefulness in data transmission remains to be evaluated on the basis of the correspondence of actual error-burst distributions with the correctable locations in the prescribed array.

Generally, this study makes appropriate reference to the preceding work. However, there is one contract research report which should be mentioned relative to this topic.[1] Kautz includes an analysis of two-dimensional parity checking as one of three extensions. These extensions are 1) row and column distances greater than two, 2) diagonal checks, and 3) multidimensional arrays. Although the principles of Calingaert's coding system were previously covered by the general theorems in Kautz's paper, the analyses of Calingaert are very important, because of the specific examples included which make it easier for engineers familiar with burst-correcting codes to utilize these extended codes. The specific formulas and tables of efficiency and probability of uncorrected error make it easier to explore the usefulness of these two-dimensional codes.

F. B. Wood
ASDD Lab.
IBM Corp.
San Jose, Calif.

[1] W. H. Kautz, "A Class of Multiple Error-Correction Codes for Data Transmission and Recording," Stanford Res. Inst., Stanford, Calif., Tech. Rept. 5, DA36-039 SC-6638; August, 1959.

R61-116 Some Further Theory of Group Codes—D. Slepian. (*Bell Sys. Tech. J.*, vol. 39, pp. 1219–1252; September, 1960.)

An earlier paper[1] by the author introduced a theory of error correcting group codes. In this paper, quite interesting additional results concerning group codes are obtained. Quoting from the author's introduction:

The main results to be found in this paper are as follows. A natural dual for a group code is defined. For any two group codes, a product code and a sum code are defined and certain properties of these operations are investigated. These operations have the important property of maintaining equivalence in the sense that $\mathfrak{A}$ *and* $\mathfrak{A}'$ *are equivalent group codes and* $\mathfrak{B}$ *and* $\mathfrak{B}'$ *are equivalent group codes, then* $\mathfrak{A}+\mathfrak{B}$ *is equivalent to* $\mathfrak{A}'+\mathfrak{B}'$ *and* $\mathfrak{A}\mathfrak{B}$ *is equivalent to* $\mathfrak{A}'\mathfrak{B}'$. *This result in turn leads to an arithmetic of equivalence classes of codes. The notion of an (additively) indecomposable equivalence class can be written in a unique manner as a sum of indecomposable equivalence classes. It is then shown that one can limit the search for best codes (with two commonly used meanings for "best") to the indecomposable equivalence classes. Enumeration formulas for the types of equivalence classes are given, and these formulas are evaluated for small values of the pertinent parameters.*

We briefly sketch some of the development. An (n, k)-*group code* $\mathfrak{A}$ is a set of 2^k n-place binary sequences that forms an Abelian group under the operation of digitwise addition modulo 2. Two (n, k)-group codes are *equivalent* if one can be obtained from the other by a permutation of the digits of each code word. An (n, k)-group code $\mathfrak{A}$ can be specified by giving any k independent code words, thus forming a $k\times n$ *generator matrix* $\Omega(\mathfrak{A})$. *Two* $k\times n$ generator matrices are equivalent, *i.e.*, generate equivalent (n, k)-group codes, if one can be obtained from the other by permuting columns and forming nonsingular linear combinations of the rows modulo 2. Every $k\times n$ generator matrix is equivalent to a matrix in M *form:* a matrix of the form $(I_k:M)$, where I_k is the $k\times k$ unit matrix and M is a $k\times(n-k)$ matrix.

Every (n, k) code $\mathfrak{A}$ has a natural *dual* $(n, n-k)$ code $\mathfrak{A}^\dagger$. If $(I_k:M)$ is a generator matrix in M form for $\mathfrak{A}$, then $(I_{n-k}:M^T)$ is a generator matrix in M form for $\mathfrak{A}^\dagger$.

The *sum* of two group codes $\mathfrak{A}$ and $\mathfrak{B}$ which are (n, k) and (n', k'), respectively, is an $(n+n', k+k')$-group code $\mathfrak{C}$ whose generator matrix is

$$\Omega(\mathfrak{C}) = \begin{bmatrix} \Omega(\mathfrak{A}) & 0 \\ 0 & \Omega(\mathfrak{B}) \end{bmatrix}.$$

$\mathfrak{A}$ is called *decomposable* if $\mathfrak{A}$ is equivalent to the sum of two or more codes, and *indecomposable* otherwise.

Theorem 2: Every (n, k) code $\mathfrak{A}$ is equivalent to the sum of indecomposable codes: $\mathfrak{A}\cong\mathfrak{A}_1+\mathfrak{A}_2+\cdots+\mathfrak{A}_m$ where $\mathfrak{A}_1$, $\mathfrak{A}_2, \cdots, \mathfrak{A}_m$ are indecomposable. Furthermore, this decomposition is unique in the following sense: If also $\mathfrak{A}\cong\mathfrak{A}_1'+\mathfrak{A}_2'+\cdots+\mathfrak{A}_{m'}'$, where $\mathfrak{A}_1', \mathfrak{A}_2', \cdots, \mathfrak{A}_{m'}'$ are indecomposable, then $m=m'$, $\mathfrak{A}_1=\mathfrak{A}_{i_1}'$, $\mathfrak{A}_2=\mathfrak{A}_{i_2}', \cdots, \mathfrak{A}_m=\mathfrak{A}_{i_m}'$, where i_1, $i_2, \cdots, i_m$ are the integers $1, 2, \cdots, m$ in some order.

The following important properties of indecomposable codes are proved:

Theorem 4: Let $\mathfrak{A}$ be a decomposable (n, k) code, $k<n$, with probability of no error $Q_1(\mathfrak{A})$. There exists an indecomposable (n, k) code $\mathfrak{P}$ whose probability of no error $Q_1(\mathfrak{P})$ satisfies $Q_1(\mathfrak{P})\geq Q_1(\mathfrak{A})$. [$Q_1(\mathfrak{A})$ is the probability that an element of $\mathfrak{A}$ will be decoded correctly when a maximum likelihood detector[1] is used as the decoder; similarly for $Q_1(\mathfrak{P})$.]

[1] D. Slepian, "A class of binary signaling alphabets," *Bell Sys. Tech. J.*, vol. 35, pp. 203–234; January, 1956.

Theorem 5: Let $\mathfrak{A}$ be an (n, k) code $k<n$ with nearest neighbor distance $d(\mathfrak{A})$. There exists an indecomposable (n, k) code $\mathfrak{P}$ with nearest neighbor distance $d(\mathfrak{P})\geq d(\mathfrak{A})$. [$d(\mathfrak{A})$ is the smallest nonzero weight (number of ones) of the elements of $\mathfrak{A}$; similarly for $d(\mathfrak{P})$.]

The *product* of two codes $\mathfrak{A}$ and $\mathfrak{B}$ which are (n, k) and (n', k'), respectively, is an (nn', kk')-group code $\mathfrak{C}$ whose generator matrix $\Omega(\mathfrak{C})$ is the Kronecker product of $\Omega(\mathfrak{A})$ and $\Omega(\mathfrak{B})$; if $\Omega(\mathfrak{A})=(a_{ij})$, then

$$\Omega(\mathfrak{C}) = \begin{bmatrix} a_{11}\Omega(\mathfrak{B}) & \cdots & a_{1n}\Omega(\mathfrak{B}) \\ a_{21}\Omega(\mathfrak{B}) & \cdots & a_{2n}\Omega(\mathfrak{B}) \\ \vdots & & \vdots \\ a_{k1}\Omega(\mathfrak{B}) & \cdots & a_{kn}\Omega(\mathfrak{B}) \end{bmatrix}.$$

Enumeration of the following quantities is described and carried out for small values of n and k:

S_{nk}, the number of equivalence classes of (n, k) codes with no zero columns;
$\bar{S}_{nk}$, the number of equivalence classes of (n, k) codes with no zero columns and no repeated columns;
R_{nk}, the number of equivalence classes of indecomposable (n, k) codes with no zero columns;
$\bar{R}_{nk}$, the number of equivalence classes of indecomposable (n, k) codes with no zero columns and no repeated columns;
W_{nk}, the number of equivalence classes of (n, k)-group codes, zero columns and repetitions allowed.

The two-part format of the earlier paper[1] is continued and makes this paper very readable: definitions, examples, and statements of important theorems in Part I, additional theorems and proofs in Part II. An interesting section entitled "Miscellaneous Comments and Problems," in which some open questions and conjectures are discussed, appears at the end of the paper.

James H. Griesmer
IBM Research Ctr.
Yorktown Heights, N. Y.

R61-117 What is a Code?—G. W. Patterson. (*Commun. Assoc. Computing Machinery*, vol. 3, pp. 315–318; May, 1960.)

After an interesting bit of etymology, the author develops, with helpful informal discussion, a mathematical definition of a code. Since the word "code" is used with a variety of different meanings, it seems worthwhile to give his definition here.

A code is a nonsingular (i.e., invertible) transformation that is defined on the elements (i.e., strings) of a syntactical base (i.e., the source language) and has for its image range a subset of another (not necessarily distinct) syntactical base (i.e., the code language). The transformation and its inverse must both be effectively calculable—that is to say, ther must exist an algorithm for coding and decoding.

The remaining half of the paper is a brief survey of some of the properties of actual codes.

D. W. Hagelbarger
Bell Telephone Labs. Inc.,
Murray Hill, N. J.

Abstracts of Current Computer Literature

(THROUGH APRIL, 1961)

These abstracts and the associated subject and author indexes were prepared on a commercial basis by Cambridge Communications Corp. under the direction of Dr. Geoffrey Knight, Jr., who also publishes the abstract journal "Solid State Abstracts," and the card services "Solid State Abstracts on Cards" and "Computer Abstracts on Cards."
—*The Editor*

CONTENTS

O. GENERAL

A Bibliography on Approximate Integration—*see* 1560.

1465
Predicting Distributions of Staff by A. Young and G. Almond (University of Liverpool); *Computer J.*, vol. 3, pp. 246–250; January, 1961.

A statistical model for predicting the number and distribution of staff among various grades in future years is described. The long-term consequences of present patterns of recruitment and promotion are revealed by a computer solution of the model.

1466
Human Factors Considerations in the Design of Electronic Computers by R. O. Lucier and E. J. Parker (University of Pennsylvania); *U. S. Govt. Res. Repts.*, vol. 35, p. 323 (A); March 10, 1961. PB 153 596 (order from LC mi$3.60, ph$9.30).

The necessity of considering the human element in the design of computer equipment, especially in the design of the operators' consoles, the use of human factors by producers of electronic computers, and the consideration of the human component of man-machine relationships in the area of ADPS are discussed. Several arguments are put forth, one of which is the need to standardize various aspects of design in the control panels of consoles. Another factor is the problem of ADP personnel—the growing need for programmers and the dwindling source of manpower, the high production of computers and the inability to find enough personnel to handle them. The appendix is concerned with various aptitude and psychological tests administered to people in ADPS. It discusses and compares the tests and the results of such tests.

1. LOGIC AND SWITCHING THEORY

Compiling Techniques for Boolean Expressions and Conditional Statements in ALGOL—*see* 1522.

1467
Linear-Input Logic by R. C. Minnick (Stanford Res. Inst.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-10, pp. 6–16; March, 1961.

Techniques for the logical design of magnetic core circuits to produce arbitrary single-output combinational switching functions are developed. The approach is based on the relationship of a single magnetic core circuit to a linearly separable switching function. A synthesis procedure which uses a pair of logical primitives, AND with NOT and OR with NOT, which are similar to the STROKE primitive and its inverse, is developed. Procedures for the synthesis of symmetric functions which require no more than the integral part of $(n+3)/2$ cores, approximately half the number used in previously published procedures, are given. The synthesis of arbitrary switching circuits is treated as a linear programming problem, and a table of all four-variable circuits in which no circuit requires more than three cores is presented.

1468
Unate Truth Functions by R. McNaughton (University of Pennsylvania); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-10, pp. 1–6; March, 1961.

Some applications of an elementary study of unate truth functions are discussed. One application is a method of deciding when a truth function is linearly separated, *i.e.*, is expressible as a linear polynomial inequality in its arguments (letting 1 represent truth and 0 represent falsity). Other applications are to contact nets and to rectifier nets.

Development of a Majority Gate for Improving Digital System Reliability—*see* 1481.

Preliminary Study of the Probabilistic Behavior of a Digital Network Majority Decision Element—*see* 1542.

1469
Axiomatic Majority-Decision Logic by M. Cohn and R. Lindaman (Sperry Rand Corp.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-10, pp. 17–21; March, 1961.

An algebra suited to logical design with majority-decision elements (parametrons, Esaki diodes, etc.) is developed axiomatically. The utility of the new algebra is demonstrated by resolving sample problems.

1470
Linear Bounded Automata by J. Myhill (University of Pennsylvania); *U. S. Govt. Res. Repts.*, vol. 35, p. 324 (A); March 10, 1961. PB 171 339 (order from OTS $0.75).

The concept of a linear bounded automaton is introduced. Such automata can do more than finite automata, but cannot do as much as Turing machines. A full discussion of the concept is given, followed by a proof that all classes of tapes of a certain kind are representable by linear bounded automata. It is also proved that all classes of tapes representable by linear bounded automata are primitive recursive.

1471
Sets of Tapes Accepted by Different Types of Automata by S. Ginsberg (System Development Corp.); *J. Assoc. Comp. Mach.*, vol. 8, pp. 81–86; January, 1961.

A number of different types of one-way automata are presented and it is shown that the family of sets of tapes accepted by at least one automaton of a particular type is the same for all types. This is accomplished by exhibiting for each automaton of each type an automaton of a previously considered type, and vice versa, such that both sets of accepted tapes are the same.

2. DIGITAL COMPUTERS AND SYSTEMS

1472
Logical Design of CG24 (A General-Purpose Computer) by G. P. Dinneen, J. A. Dumanian, *et al.* (Lincoln Lab., M.I.T.); *U. S. Govt. Res. Repts.*, vol. 35, p. 190 (A); February 10, 1961. PB 148 281 (order from LC mi$3.30, ph$7.80).

A detailed design for a high-speed general-purpose digital computer is presented. The design considerations are governed by the assumption that implementation of the machine is to be accomplished using only solid-state devices. Sections I–V describe the essential characteristics, structure and method of design of the computer. Sections VI–IX discuss its detailed logical structure.

1473
Computer Design of Multiple-Output Logical Networks by T. C. Bartee (Lincoln Lab., M.I.T.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-10, pp. 21–30; March, 1961.

An important step in the design of digital machines lies in the derivation of the Boolean expressions which describe the combinational logical networks in the system. Emphasis is generally placed upon deriving expressions which are minimal according to some criteria. A computer program which automatically derives a set of minimal Boolean expressions describing a given logical network with multiple-output lines is discussed. The program accepts punched cards listing the in-out relations for the network, and then prints a list of expressions which are minimal according to a selected one of three criteria. The basic design procedure and the criteria for minimality are described.

1474
Computers as an Aid in Computer Design Assessment by J. M. Bennett and R. J. Dakin (University of Sydney); *Computer J.*, vol. 3, pp. 253–255; January, 1961.

Simulation routines for assessing the utility of various design proposals for a new computer, SNOCOM, to be run on the existing SILLIAC computer, are described. By this means, rational design decisions can be made without any hardware expenditure. The preparation of a realistic program mix for assessing computer performance is also discussed.

1475
The Use of Index Calculus and Mersenne Primes for the Design of a High-Speed Multiplier by A. S. Fraenkel (University of California); *J. Assoc. Comp. Mach.*, vol. 8, pp. 87–96; January, 1961.

A system of multiplication in which the operands are represented by their residues modulo a set of Mersenne primes, and the residues are then represented as powers of a primitive root, is described. Multiplication may then be replaced by adding the indexes of the powers of the primitive roots within each residue class. Means of computing the indexes from the number and the number from the indexes are presented. The difficulties of division in the system, except for special cases, are indicated.

1476
A Digital Correlator Based on the Residue Number System by P. W. Cheney (Lockheed Aircraft Corp.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-10, pp. 63–70; March, 1961.

A system design for a digital correlator based on the application of the residue number system for computation is presented. Areas of investigation include sampling, ana-

log-to-residue conversion, logical design of the arithmetic units, residue-to-analog conversion, and modes of operation of the proposed digital correlator. The advantages of speed of computation and simplicity of logic due to the use of a residue number system are shown to result in a significantly faster and simpler system than if a conventional number system were used. The resulting digital correlator is designed for megacycle sampling and computation with a 0.1 per cent system precision.

1477
Multiple Precision Division by P. Rabinowitz (Weizman Inst. of Science); *Commun. Assoc. Comp. Mach.*, vol. 4, p. 98; February, 1961.

An extension of the Newton division algorithm that reveals its relation to a truncated series expansion is described. A multiple precision algorithm for reciprocals $x_{i+1}=x_i+x_i(1-Nx_i)+x_i(1-Nx_i)^2+\cdots$. results. This device can be extended to an arbitrary power of N, by expressing it in the form $N^p=x_0[1-(Nx_0^{-1/p})]^p$, and expanding binomially.

1478
Optimum Time for Multiplication on a Digital Computer by H. H. Johnson (Inst. of Tech., Bradford); *Computer J.*, vol. 3, pp. 256–261; January, 1961.

Four methods of multiplication for serial binary computers with a cyclic main store are discussed. They are the pencil-and-paper method of adding in successive partial products, adding the multiplicand at the end of a sequence of zero multiplier digits and subtracting it at the end of a sequence of ones, and two modifications of the second method that improve its performance. Substantial savings in multiply times are obtained, particularly with the last two modifications. It is shown that very little is gained when multiplicand and multiplier are inspected to see which has an optimal sequence of ones and zeros.

1479
A General Method of Applying Error Correction to Synchronous Digital Systems by D. B. Armstrong (Bell Telephone Labs.); *Bell Sys. Tech. J.*, vol. 40, pp. 577–593; March, 1961.

A general method for applying error correction to a synchronous digital system, only one unit of which is presumed to be malfunctioning at any given time, is described. The method of triplication and "vote-taking" is included as a special case. In principle, the sytem may operate continuously even when a fault is present, or during maintenance. Percentage redundancy decreases as system complexity increases, but triplication of equipment even for moderately complex systems may be required. Some specialized error-correcting schemes and some methods of estimating redundancy and reliability improvement are presented.

Margin Considerations for an Esaki Diode-Resistor OR Gate—*see* 1485.

1480
Transistor Life in the TX-O Computer After 10,000 Hours of Operation by D. J. Eckl and R. L. Burke (Lincoln Lab., M.I.T.); *U. S. Govt. Res. Repts.*, vol. 35, p. 50 (A); January 13, 1961. PB 150 601 (order from LC mi$3.30, ph$7.80).

The results of a test of 58 microalloy transistors and 800 surface-barrier transistors are presented after 10,000 hours of operation in a computer. During the entire period of operation, only one transistor—a light-pen-amplifier—was removed as an "uncaused" failure. No data are available on "personnel-induced" failures. Approximately 10 per cent of the transistors tested fell below original acceptance specifications, but they were returned to service and have operated normally. Certain of the parameter variations were unexpected and appear to be characteristic of surface changes in the surface-barrier transistor. The results of the transistor tests so far are excellent and surpass those originally anticipated.

1481
Development of a Majority Gate for Improving Digital System Reliability by R. Wasserman (Hermes Electronics); *U. S. Govt. Res. Repts.*, vol. 35, p. 81 (A); January 13, 1961. PB 150 484 (order from LC mi$3.60, ph$9.30).

Logical design procedure for the efficient use of redundancy in improving the reliability of digital systems is discussed. This calls for the development of a highly reliable, simply constructed, majority gate. Requirements and design considerations for the majority gate using magnetic cores as logical building blocks are presented. Logical design principles for a shift register, dynamic flip-flop, logical gate functions, binary counter, comparator, and two-input full adder are also considered. A 6-core majority gate module and a 9-core majority gate module are discussed.

1482
The Backboard Wiring Problem: A Placement Algorithm by L. Steinberg (Remington Rand); *SIAM Rev.*, vol. 3, pp. 37–50; January, 1961.

A mathematical form of a backboard wiring problem, the placement of computer elements on the backboard, is presented, and a partial solution of the problem is suggested. An algorithm is developed using the techniques of linear programming (Munkres' solution of the assignment problem) and applied to a sample problem. Two norms are defined, a weighted-wire-length norm and a maximum-wire-length norm; it is shown that the latter can be regarded as a limiting case of the former.

1483
Orion; *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 110–113; February, 1961.

The basic organization and specifications of the Ferranti Orion Computer are described. Special attention is paid to the time sharing of the central processor by several programs, and the priority scanning circuitry which controls it. This feature mainly benefits commercial users with a high rate of input-output activity. For scientific programs, the main advantage is that debugging can take place simultaneously with a production run. The machine has an addition time of about 50 μsec and is said to be equally satisfactory for scientific and commercial use.

1484
A Semiconductor Binary Coordinate Converter by M. F. Williams, A. F. Thornhill, and W. A. Richards (U. S. Naval Res. Lab.); *U. S. Govt. Res. Repts.*, vol. 35, p. 324 (A); March 10, 1961. PB 149 297 (order from LC mi$2.40, ph$3.30).

The characteristics of a semiconductor binary digital coordinate converter are discussed. This equipment converts 18-bit polar-coordinate data to 20-bit rectangular-coordinate data in 3 msec. The equations used are: $X=\rho \sin\theta$ and $Y=\rho\cos\theta$. Sine θ and cosine θ are obtained by the use of a rectangular diode matrix and diode trigonometric function tables; X and Y are obtained by summing partial products in a parallel-accumulator-type multiplier. The converter occupies a volume of less than two cubic feet and requires 1 ampere at 20 volts and 200 ma at 90 volts. Methods of increasing the operating rate of the converter are considered.

3. DEVICES AND BASIC LOGIC AND WAVEFORMING CIRCUITS

1485
Margin Considerations for an Esaki Diode-Resistor OR Gate by H. K. Gummel and F. M. Smits (Bell Telephone Labs.); *Bell Sys. Tech. J.*, vol. 40, pp. 213–232; January, 1961.

An Esaki diode-resistor logic circuit powered from a three-phase supply and involving OR gates is analyzed. Practical switching times are of the order of 10 $|R-C|$. The voltages at which the current maximum and the current minimum occur set an upper limit on the achievable logical gain. For a sum of fan-in plus fan-out of 3, the margins on key diode and circuit parameters must be better than ± 2 per cent, with all margins assumed equal. The margins can be ± 3.5 per cent for a fan-in plus fan-out of 2, which, however, restricts the applications to shift registers, flip-flops, and the like.

1486
Bilateral Switching Using Nonsymmetric Elements by M. Aoki and G. Estrin (University of California); IRE Trans. on Electronic Computers, vol. EC-10, pp. 42–50; March, 1961.

Magnetic-core memory elements characteristically require bipolar applied fields. The vanishing inner diameter of toroids and the loss of the third dimension entirely in deposited thin films demands minimization of the number of wires. A configuration which has been investigated and applied in a word organized memory is illustrated. It consists of a pair of mutually inverted and parallel connected transistors that are not in general symmetrical. Some of the system considerations which determine the important design parameters are discussed. Methods for location of regions of satisfactory

operation in the many-variable space of the inverted transistor pair are described. Although a particular design problem is discussed, attention is focused on the question, "what classical and new procedures can we use to reduce the number of dimensions in such design problems?" The power of the computer as a design tool is crucially dependent upon such processes.

1487
An Integrated Binary Adder by M. E. Szekely, S. M. Marcus, and J. T. Wallmark (RCA Res. Ctr.); *U. S. Govt. Res. Repts.*, vol. 35, p. 81 (A); January 13, 1961. PB 152 045 (order from LC mi$2.40, ph$3.30).

A integrated binary full adder which uses unipolar field-effect transistors as its active and passive elements is described. A binary full adder is a computer circuit that adds three binary numbers, X, Y and a carry from a previous stage, C, giving two outputs, one for the sum and one for the carry to the next stage. Each one of these binary numbers may be 0 or 1 corresponding to eight possible combinations. The design is developed in terms of conventional circuitry, and then extended to an integrated module. The active element used throughout the adder is the unipolar field-effect transistor (UT), the operation of which is described. The only passive elements used are resistors which are also in the form of unipolar transistors. The logic concept used is direct-coupled unipolar transistor logic (DCUTL), which is also described.

1488
Inertial Selection for Magnetic Core Logic by R. H. Tancrell (Lincoln Lab., M.I.T.); *U. S. Govt. Res. Repts.*, vol. 35, p. 190 (A); February 10, 1961. PB 152 637 (order from LC mi$2.70, ph$4.80).

The Inertial Selection scheme, in which logical functions can be performed by connecting toroidal ferrite cores together with single-turn windings only, is described. A "flux gain" feature is thereby obtained which allows more flux to be switched into a receiving core than is received there from a transmitting core. To achieve these characteristics, the Inertial Selection scheme utilizes two thresholds of low-coercive-force ferrite cores. It also uses the property that one of these thresholds can be changed by electrical means. The factors involved in understanding these magnetic characteristics and their utilization in logic applications are discussed in detail.

1489
Impulse Selection for Core Logic by R. H. Tancrell (Lincoln Lab., M.I.T.); *J. Appl. Phys.*, suppl. to vol. 32, pp. 40S-41S; March, 1961.

Impulse selection, a scheme for operating ferrite cores which have low values of coercive force, is discussed. Two different threshold properties of a core are utilized. One threshold is the dc coercive force, which can be changed electrically. The other threshold results from the inertial magnetic effects within the core which are predominant when pulses of very narrow widths are applied. The switching behavior of ferrite cores when two threshold currents, a low-amplitude current and an impulse of current, are combined, and the applications of this property to core logic are considered.

1490
Analysis of Ferrite Core Switching for Practical Applications by P. A. Neeteson (N. V. Philips); *Electronic Applications*, vol. 20, pp. 133–152; 1959–1960.

Analysis of the behavior of magnetic switching materials having rectangular hysteresis loop properties, based as closely as possible on the physical background of the flux reversal process, necessarily leads to expressions which are too complex to handle in the practical design of circuits in which the core properties are used. On the other hand, a rough approximation may render possible design procedures leading to quick results, but experimental corrections may be required at a later stage. A solution of this problem, in which the derived analytical expressions are in close agreement with the physical behavior of the material and are at the same time fairly easy to apply to circuit design, is presented. Resistive and inductive loading of the cores are discussed, and also cores driving a line of cores in a memory stack, current sources being used to drive the cores. The operation of voltage-driven cores is also dealt with.

1491
Ferrite Toroid Core Circuit Analysis by R. Betts and G. Bishop (IBM Corp.); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-10, pp. 51–56; March, 1961.

An analysis of the terminal characteristics of thin ferrite toroid cores under arbitrary drive and load conditions is presented. The analysis is founded only on the following two experimentally confirmed conditions: 1) the time required for a complete reversal of flux under unloaded conditions is inversely proportional to the magnitude of a step-driving field which is in excess of the critical field required to initiate flux change; and 2) the open circuit voltage-time output waveforms caused by step-driving currents are identical when normalized with respect to amplitude and time. The normalized output voltage waveform $f'(x)$ is used to develop a terminal characteristic equation. It is shown that $f'(x)$ may be obtained by using a nonideal step-input current. Utilizing a modified Gaussian equation to represent $f'(x)$, equations are developed to allow the prediction of core response to arbitrary input waveforms, using four parameters easily obtained from voltage response vs NI step-drive plots, and $f(x)$, which is the integral of the normalized expression for the open-circuit voltage $f'(x)$ and is proportional to the flux switched in the core. The equations are expanded to include a load circuit and to test the validity of the expressions developed. Theoretical and experimental results are compared for a core loaded with series RL and RLC circuits with both ramp and step-drive curents. Agreement is shown to be good, even though the core used was not particularly thin.

1492
Possibilities of All-Magnetic Logic by U. F. Gianola (Bell Telephone Labs.); *J. Appl. Phys.*, suppl. to vol. 32, pp. 27S-34S; March, 1961.

A sequential logic circuit must have a capacity for memory, undirectionality, and gain. The problem of obtaining these features in magnetic components is considered, and several approaches are described. The organization of a circuit using multiapertured cores is discussed, and a method for introducing added combinational logic in a compatible manner is described. The status of these techniques is summarized.

1493
Magnetic Film Devices Using Passive Loading by J. M. Daughton, T. A. Smay, A. V. Pohm, and A. A. Read (Iowa State University); *J. Appl. Phys.*, suppl. to vol. 32, pp. 36S-37S; March, 1961.

Computer solutions of a pair of differential equations based on the Landau-Lifshitz model, which indicate that with passive loading, magnetic film devices may be used as storage and logical elements in digital computers, are reported. Experimental models of a flip-flop consisting of an RCL loaded film device show that successive 20-oe, 0.1-μsec drive field pulses can switch the magnetization in the film alternately from one rest orientation to the other with repetition rates of the drive on the order of 10^6 pps. A nondestructive readout memory element consisting of an RL loaded thin film device is shown which allows films with H_k on the order of 3 oe to be driven in the hard direction by 15-oe, 0.05-μsec field pulses without destroying the stored information, resulting in output voltages of approximately 25 mv per winding turn. Fabrication of a memory employing such elements by evaporation techniques is discussed.

1494
Computer-Device Applications of Thin Ferromagnetic Films (Sperry Rand Corp.); *U. S. Govt. Res. Repts.*, vol. 35, pp. 82–83 (A); January 13, 1961. PB 150 767 (order from LC mi$2.70, ph$4.80).

The results of a study of the application of thin ferromagnetic films to computers are reported. Anisotropic, "double threshold," and "rotating anisotropy" films have been considered for use as logic and memory devices. A plausible explanation of the unique properties of double threshold and rotating anisotropy films is given in terms of domain walls and impurity atom diffusion. Two NDRO memory devices using double threshold and rotating anisotropy films are described. An AND gate, an EXCLUSIVE OR gate, and three types of full adder using thin films are described. An AND gate, an EXCLUSIVE OR gate, and a NDRO memory element using double threshold films have been constructed and evaluated.

1495
A Multi-Purpose Computer Element by C. B. Taylor (E.M.I. Electronics Ltd.); *Electronic Engrg.*, vol. 33, pp. 96–99; February, 1961.

A long-tailed pair waveform reshaper which, together with a number of logical gates, has many uses as a basic element for a computing system is described. The main design aim was to produce an economical binary counter stage employing transistor-diode logic, but among the other possible uses of the element are a pulse generator, simple reshaper, delay with reshaped bi-phase outputs bistable element, and free-running square-wave generator.

4. STORAGE AND INPUT-OUTPUT

1496
A Magnetic Associative Memory by W. L. McDermid and H. E. Petersen (IBM Corp.) *IBM J. Res. and Dev.*, vol. 5, pp. 59–62; January, 1961.

The construction of an experimental magnetic core associative memory that is also capable of being interrogated conventionally is described. The method of nondestructive readout and the construction of suitable detectors to read the available signal energy are discussed. Memory capacities of several hundred words appear feasible with a single interrogation driver. A search and retrieval time of 6 μsec on a 36-bit search basis has been achieved.

1497
A Digital Static Magnetic Wire Storage with Nondestructive Read-Out by C. G. Shook (Genl. Dynamics); IRE Trans. on Electronic Computers, vol. EC-10, pp. 56–62; March, 1961.

After a brief review of pertinent magnetic effects and sonic wave propagation in elastic media, a nonvolatile, digital, magnetic storage scheme is described, wherein binary words may be stored by magnetizing segments of a wire, and the information may be read out an unlimited number of times with no deterioration of the stored information. Two storage schemes are presented: a temporary, electrically addressed storage, and a permanent, program-type store. Bit-storage density, read-out and input pulse shapes, and read-out frequency are noted. Possible limitations such as losses, temperature effects, and pulse shape are balanced against advantages, and a comparison with a number of other types of bit storage is made.

Computer-Device Applications of Thin Ferromagnetic Films—*see* 1494.

1498
Research on Magnetic Rod Storage and Switching Device by A. J. Kolk (Natl. Cash Register Co.); *U. S. Govt. Res. Repts.*, vol. 35, p. 207 (A); February 10, 1961. PB 152 520 (order from LC mi$5.70, ph$16.80).

The results of a study of the magnetic properties of thin film, iron-nickel alloys electrodeposited in a rod configuration are presented. These properties were found to be critically influenced by the physical condition of the conducting substrate surface. Silvered-glass and beryllium-copper substrates were subjected to intensive investigation. The effects of varying other plating parameters such as current density, pH, bath composition, temperature, organic additives, plating time, and the presence of an axial magnetic field were also studied. Electron micrographs were made of the substrate surface and the plate at different plate thicknesses. An attempt was made to correlate residual stress measurements with the properties of the plate. Preferred orientation in the plate was investigated by means of X-ray diffraction techniques. Finally, a series of experimental switching curves was determined and a mechanism for the switching phenomena was proposed.

1499
Demagnetization of Twistor Bits by W. A. Barrett (Bell Telephone Labs.); *J. Appl. Phys.*, suppl. to vol. 32, pp. 35S-36S; March, 1961.

A search-coil technique for measuring the effect of demagnetizing fields on twistor wire is described. A coincident-current bit select followed by a single-current read select is used to form and to "erase" the magnetized regions. It is found that the shape of the magnetized regions thus formed is a complex function of the two currents used and that the bits are not fully erased by the single current. The results suggest that demagnetization must be reduced by reducing thickness and/or bit density unless a more complex solenoid structure can be found to control the magnetization of the bit regions.

1500
Card Capacitor—A Semipermanent, Read Only Memory by H. R. Foglia, W. L. McDermid and H. E. Petersen (IBM Corp.) *IBM J. Res. and Dev.*, vol. 5, pp. 67–68; January, 1961.

The construction of a card-capacitor read-only store is described. The information to be stored is punched on a metallized IBM card, with a hole representing a stored one. The card is inserted between two printed-circuit boards, the sense lines on one being placed orthogonally to the read lines on the other, and the foil portion of the card is grounded. Up to 500 cards may be stacked together. Memory systems, ranging from 10^7 bits at 10μsec to 10^4 bits at 10^{-1} μsec access time are feasible. Power requirements are low and information is easily altered by replacing cards.

1501
Selective Erasure and Nonstorage Writing in Direct-View Halftone Storage Tubes by H. N. Lehrer (Hughes Res. Labs.); Proc. IRE, vol. 49, pp. 567–573; March, 1961.

A new type of direct-view halftone storage tube that can selectively erase as well as simultaneously display stored and nonstored information is described. A dual-effects target is used; one effect (secondary emission) charges the storage surface positively, while the other (bombardment-induced conductivity) charges it toward the backing-electrode potential, which in this case is negative. Selection of the effect and, consequently, the charging direction, is determined by the incident beam energy. At low energies, secondary emission predominates and the target is written on, *i.e.*, charged positively. At high energies, the bombardment-induced conductivity prevails and the target is erased, *i.e.*, the positively charged areas are discharged. The two effects cancel at an appropriate intermediate beam energy, thus permitting presentation of nonstored information without otherwise disturbing the display, by means of that portion of the beam current which passes through the backing electrode and strikes the viewing screen. A 5-inch, direct-view halftone storage tube that utilizes such a dual-effects target is described and its characteristics are discussed.

1502
Experiments on Magnetic Tape Readout with an Electron Beam by M. M. Fruendlich and D. I. Breitzer (Airborne Instruments Lab.), S. J. Begun and J. B. Gehman (Thompson Ramo Wooldridge), and J. K. Lewis (Dept. of Defense); Proc. IRE, vol. 49, pp. 498–509; February, 1961.

A readout system in which a magnetic tape is bent over a cylinder and an electron beam is made to pass across the recorded track is described. The electrons close to the tape form a cycloid pattern which is deflected across the entrance slit of a Faraday collector, thereby producing an output signal corresponding to the recorded information. At present, some practical limitations prevent the full realization of the system's potential. The ultimate limitation is the shot noise of the electrons. Theoretical considerations indicate that a 3-Mc-wide video signal, for example, if suitably recorded, could be reproduced with a signal-to-shot-noise ratio of 30 db and with an area packing density of 0.85 cycle per square mil.

1503
Recording and Reproduction of NRZI Signals by R. S. Schools (IBM Corp.); *J. Appl. Phys.*, suppl. to vol. 32, pp. 42S-43S; March, 1961.

A theoretical study of a saturation type of recording and reproduction in digital magnetic tape systems is described. The dependence of tape magnetization and of output signal pulse width and amplitude on the major system parameters has been investigated and determined. These parameters are the magnetic field distribution of the recording head, write current waveform, tape thickness and hysteresis characteristics, head-to-tape spacing, reading gap length, and bit density. A portion of this study concerns the extension of the sinusoidal theory to include the reproduction of signals resulting from saturation recording. This is done by a Fourier analysis of the tape magnetization and a term-by-term summation of harmonic responses to obtain the resultant signal. The results show why high writing field gradients, high tape B-H "squareness," and thin tapes are needed for high-density recording. They indicate possible reasons for EMF pulse peak position asymmetries during readback. Two examples of output calculations are included. One shows the signal amplitude and pulse width vs bit density, compared with experimental measurements. The other presents the field distribution above the magnetized tape, indicating the influence of the presence of the read head.

Use of Magnetic Tape for Data Storage in the ORACLE-ALGOL Translator—*see* 1511.

Human Factors Considerations in the Design of Electronic Computers—*see* 1466.

1504
The Calliscope: A Versatile Alphanumeric Display by K. E. Perry and E. J. Aho (Lincoln Lab., M.I.T.) *U. S. Govt. Res. Repts.*, vol. 35, p. 82 (A); January 13, 1961. PB 150 598 (order from LC mi $2.40, ph $3.30).

The calliscope, a digitally controlled alphanumeric and graph display intended for presentation of output data from a digital computer, is described. This display employs a magnetically focused and deflected cathode-ray tube. Spot positioning is accomplished by digital switching of high currents through a low-inductance yoke. Settling time is 20 μsec. Alphanumeric symbols are generated by a calligraphic or spot writing method which synthesizes the X and Y current waveforms necessary to control the motion of the spot.

1505
The TX-2 Electrostatic Display System by R. E. Savell (Lincoln Lab., M.I.T.); *U. S. Govt. Res. Repts.*, vol. 35, p. 190 (A); February 10, 1961. PB 152 468 (order from LC mi$2.40, ph$3.30).

A display that is produced in a 7-inch square on the face of an electrostatically deflected cathode-ray tube is described. Ten bits are used in each axis to address the location of the spot to be displayed. The bits are decoded to produce the deflection voltages, and after a 7-μsec setup time, the spot is intensified for 10, 20, 40 or 80 μsec according to instructions received from the computer. At present, no focus correction is used with the system; the output of the focus-correction amplifier is tied to ground. Work is in progress on a focus-correction circuit and on modifications to improve decoder stability. The display is used for visual observation, photographic observation using Polaroid, 16-mm or 35-mm cameras, or photoelectric observation. In the photoelectric application a photodiode mounted in a "light pen" is used to sense the display output. The light pen (see abstract 727) output can then be used to modify the computer program. This system has also been used as a flying-spot scanner, with a photomultiplier replacing the light pen, to provide photographic and character-recognition inputs to TX-2.

5. PROGRAMMING AND CODING

1506
The Evolution of Programming Systems by W. Orchard-Hays (CEIR); Proc. IRE, vol. 49, pp. 283–295; January, 1961.

A comprehensive history and summary of the present state of the art of programming systems, ranging from the simplest machine-oriented assembly programs through compilers, processing, data storage and retrieval systems and list processors to the highest levels of abstraction, are presented. The tremendous influence of programming practices on the design of computer systems is detailed, emerging programming techniques and principles are reviewed, and the necessity for relieving reprogramming problems and for the integration of computing facilities efficiently with input-output operations is stressed.

Applications of Graphs and Boolean Matrices to Computer Programming—*see* 1571.

1507
Survey of Modern Programming Techniques by R. W. Bemer (IBM Corp.); *Computer Bull.*, vol. 4, pp. 127–135; March, 1961.

Modern techniques in the field of programming, including basic language elements, machine dependent and machine independent languages, analysis languages, processor techniques, and finally whole operating systems are reviewed. The importance of symbol manipulation techniques as the key to the ability to write a programming system in its own language, and the versatility resulting from this ability are stressed. The development of universal programming languages such as ALGOL and COBOL, the importance of bootstrapping techniques, scheduling, and the efficient interleaving of input-output and computing operations are discussed.

1508
Computational Chains by T. Marill and T. G. Evans (Bolt, Beranek and Newman, Inc.), *U. S. Govt. Res. Repts.*, vol. 35, p. 189 (A); February 10, 1961. PB 151 519 (Order from LC mi $2.70, ph $4.80).

A formalism within which questions relating to the efficiency of information-manipulative processes may be studied is developed. An example of the application of this theory to the simplification of a computer program is given and the direction of future work is indicated.

1509
Considerations in Choosing a Character Code for Computers and Punched Tapes by H. McG. Ross (Ferranti Ltd.); *Computer J.*, vol. 3, pp. 202–210; January, 1961.

The choice of a suitable commercial data-oriented code for alphanumeric characters is discussed. A basic 6-bit code is obtained from which 7- and 8-track punched tape codes are derived. The system is realizable with minor variants on existing equipment and permits future development in a variety of directions, including a compatible code for extended alphabets. Particular features are an Escape character to indicate that the following characters are not given their usual code interpretation, and shift-in and shift-out characters.

1510
The BKS System for the Philco-2000 by R. B. Smith (Westinghouse Electric Corp.); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 104 and 109; February, 1961.

A system that controls the loading of programs from cards or master tape, provides common subroutines, assigns program tapes to tape transports, and directs all mounting and removing of tapes is described. Certain restrictions impose some uniformity on the individual programs controlled by the system.

1511
Use of Magnetic Tape for Data Storage in the ORACLE-ALGOL Translator by H. Bottenbruch (Oak Ridge Natl. Lab.); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 15–19; January, 1961.

The use of magnetic tapes as auxiliary storage for an ALGOL translator on a machine with limited internal memory (2000 words) is described in detail. Extensions of the technique for the tape storage of programs too large to fit unsegmented into the internal memory are discussed.

1512
The Internal Organization of the MAD Translator by B. W. Arden, B. A. Galler, and R. M. Graham (University of Michigan); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 28–31; January, 1961.

The internal structure of a translator for MAD, a language similar to ALGOL, is described. The translator has been designed for high-speed translation, generality, and object program efficiency. The over-all procedure is subdivided into Statement Decomposition, Storage Allocation, and Generation of the Object Program. Each phase is described in some detail.

1513
The SLANG System by R. A. Sibley (IBM Corp.); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 75–84; January, 1961.

The development of a machine-independent language (SLANG) patterned after ALGOL 58 to facilitate the construction of compilers is described. Results show compilation manner, and that the description can be translated into the machine language of any of a class of computers by a single program running on a single computer. The system accepts as inputs a set of statements A, describing the compilation process for some problem-oriented language L and a set of statements B describing the computer C upon which the compilation process for L is to take place. The system will produce as outputs a machine-language program for C capable of carrying out the processes described by A, and also appropriate documentation. Problem areas relating to the efficiency of the machine code produced and the completeness of statements B await investigation.

1514
The CLIP Translator by D. Englund and E. Clark (System Development Corp.); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 19–21; January, 1961.

The CLIP translator, which translates statements from CLIP, a language based on an augmented ALGOL language, to a symbolic IBM 709 language, is described. Extensive use is made of generated tables and strings in the translation procedure. CLIP has been written in its own language and has successfully reproduced itself. It is also being used to write the JOVIAL family of compilers in an effort to make the generation of compilers completely machine independent.

1515
Recursive Processes and ALGOL Translation by A. A. Grau (Oak Ridge Natl. Lab.); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 10–15; January, 1961.

A "push-down" technique for mirroring the recursive nature of many ALGOL entities in an ALGOL translator is described. Subroutines that can slave themselves on a new level without losing information on the old are handled by the "push-down" technique. Theoretically, one push-down list is sufficient, but in practice it is desirable to use several. Detailed definitions of the main terms required are presented.

1516
Comments on the Implementation of Recursive Procedures and Blocks in ALGOL 60 by E. T. Irons (Princeton University) and W. Feurzeig (University of Chicago); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 65–69; January, 1961.

Methods for implementing the recursive procedures inherent in ALGOL 60, in which subroutines call themselves at a lower depth, are described. The use of a push-down storage for local parameters is discussed. Because of possible difficulties in detecting recursion at compile time, it is detected and handled dynamically at run time. In this way, the implementation of procedures or blocks not containing recursion is not slowed down.

1517
Running Pegasus Autocode Programs on Mercury by A. Gibbons (University of Manchester); *Computer J.*, vol. 3, pp. 232–236; January, 1961.

A new general method of translating autocodes is applied to the translation of Pegasus Autocode into Mercury machine code. In addition to describing the translation scheme, details connected with simulating the Pegasus Autocode on the Mercury machine are discussed. By this means, Pegasus problems may be run on Mercury at up to 25 times their original speed.

1518
A Syntax Directed Compiler for ALGOL 60 by E. T. Irons (Princeton University); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 51–55; January, 1961.

Most compilers serve not only to translate an object language into a target language, but also to define the object language in terms of the target language. A compiling system which separates the functions of definition from that of translation is described. First a meta-language (an extension of the meta-language of the ALGOL 60 report), which is used to define the object language, is presented. A program which uses a direct machine representation of the meta-linguistic specifications to effect a translation is then described. The system outlined has been implemented to translate the entire ALGOL 60 language into a CDC-1604 assembly language.

1519
CL-1, An Environment for a Compiler by T. E. Cheatham, Jr., G. O. Collins, Jr., and G. F. Leonard (Technical Operations Inc.); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 23–28; January, 1961.

A flexible, large-scale programming system to facilitate the solution of information processing problems and to provide intercommunication between programs and programmers is described. The system is based on a master file concept and has provisions for accepting, storing, and retrieving both descriptions and instances of large and complex data sets, as well as algorithms defined on these data sets. Both data and algorithms may be expressed in a family of command and descriptive languages. The concept of distinct data descriptions and the content and use of such descriptions are discussed in some detail.

GENDARE System: Fundamental Concepts of Logic and Structure—*see* 1625.

1520
Computer Languages for Symbol Manipulation by B. F. Green, Jr. (Lincoln Lab., M.I.T.); IRE Trans. on Human Factors in Electronics, vol. HFE-2, pp. 3-8; March, 1961.

Complex flexible computer programs written easily in list-processing languages are discussed. Storage registers are linked in arbitrary sequences to form lists and list structures; special provisions are made for recursive subroutines and for hierarchical programs.

1521
THUNKS, a Way of Compiling Procedure Statements with some Comments on Procedure Declarations by P. Z. Ingerman (University of Pennsylvania); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 55–58; January, 1961.

The basic problem in the compilation of procedure statements and declarations is that of transmission of information such as the value of a parameter, the location where a value is to be stored, and the location to which a transfer is to be made. In these cases the provision of an address answers the requirements. A THUNK is a piece of coding which when executed provides an appropriate address in a fixed location, such as an accumulator or index register. The syntax of the formation of the actual parameter enables the compiler to distinguish the level of address supplied by the THUNK.

1522
Compiling Techniques for Boolean Expressions and Conditional Statements in ALGOL by H. D. Huskey (University of California) and W. H. Wattenburg (Bendix Corp.); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 70–75; January, 1961.

A compiling system that translates conditional statements and Boolean assignment statements as defined in ALGOL 60 into a suitable intermediate language is described. The target program has a True and a False exit, depending on the truth value of the object Boolean expression. Efficiency is obtained by not testing further variables of a disjunction if one of them has the value True, and *mutatis mutandis* for conjunctions.

1523
Allocation of Storage for Arrays in ALGOL 60 by K. Sattley (University of Chicago); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 60–65; January, 1961.

A programming mechanism for dynamically determining the location and size of a variable-sized array at the time of running a program is described. Associated with each array is a "dope vector" giving complete information about the current parameters of the array. Storage is allocated for program, constants, etc., at the low end of memory. The rest of memory is available to a special allocator program called FUSBUDGET. Up to the point where no more memory is available, storage for variable arrays is accomplished efficiently at little cost in time or space.

1524
Dynamic Declarations by P. Z. Ingerman (University of Pennsylvania); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 59–60; January, 1961.

A procedure for mapping one array into another when the subscript bounds may change dynamically is described. A dope vector giving all the essential information about the subscripts appearing in the array is associated with every array. If the array itself is stored in numerical order by subscript, it can be represented by an equivalent vector. The mapping of this equivalent vector into a new vector representing the augmented or diminished array is described.

1525
Two Systems for Symbol Manipulation with an Algebraic Compiler by J. W. Carr, III and J. W. Hanson (University of North Carolina); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 102–103; February, 1961.

Two simple symbol decomposition and recomposition routines which, when combined with the University of North Carolina version of the IT compiler or the University of Michigan GAT compiler, give satisfactory manipulation facilities are described. The inclusion of the subroutines allows the decomposition of input words into a sequence of one sequence words and the inverse operation of recomposition.

1526
Computer Generation of Optimized Subroutines by H. H. Denman (Wayne State University); *J. Assoc. Comp. Mach.*, vol. 8, pp. 104–118; January, 1961.

A method for the computer generation of function-evaluation subroutines is described. The approximation formula used is nearly the optimal polynomial approximation (in the Tchebycheff sense) for the function in the given interval. Some results generated for a set of common functions are presented.

1527
A Basic Compiler for Arithmetic Expressions by H. D. Huskey (University of California) and W. H. Wattenburg (Bendix Corp.); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 3–9; January, 1961.

A basic compiler for arithmetic expressions that will compile intermediate language programs from assignment statements and arithmetic expressions is described. Object programs are compiled in a single left to right scan of the source language statements. A push-down technique for storing a temporary command list is used. The object programs generated are optimum in the sense that a minimum number of storage accesses are used. A Fortran version of the compiler is given in an appendix.

1528
An Algorithm for Coding Efficient Arithmetic Operations by R. W. Floyd (Illinois Inst. Tech.); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 42–51; January, 1961.

It is shown that more efficient machine code sequences are obtained by scanning arithmetic statements right to left instead of left to right. Fewer fetch and store operations result, constant subexpressions are evaluated during compilation and many equivalent subexpressions are recognized. Detailed flow charts and ALGOL encoded examples are presented.

1529
MADCAP: A Scientific Compiler for a Displayed Formula Textbook Language by M. B. Wells (University of California); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 31–36; January, 1961.

MADCAP, a scientific compiler that can accept source statements written in conventional mathematical notation and in easily typed form is described. MADCAP linearizes its source programs before translation; the MADCAP language also maintains a close intuitive connection with the language of flow charts. Time-consuming analyses and optimizations are avoided on the basis that many of the programs translated will be of a temporary nature.

1530
The Use of Threaded Lists in Constructing a Combined ALGOL and Machine-Like Assembly Processor by A. Evans, Jr., A. J. Perlis, and H. Van Zoeren (Carnegie Inst. Tech.); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 36–41; January, 1961.

Generation of a threaded list procedure for formula translation which is free of recursive processes is described. Translation of linear source statements into a tree format is utilized. The slowness of the resulting procedure is compensated by its flexibility, ease of optimization, and natural representation of control processes. The inclusion of list processing hardware in computers is suggested as a means of improving speed.

1531
Multiple Program Data Processing by B. L. Ryle (Marc Shiowitz and Assoc.); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 99–101; February, 1961.

Three types of multiple program processing, *i.e.*, Central Memory-Multiple Control, Multiple Memory-Central Control, and Multiple Memory-Multiple Control, are distinguished. One system of each type, the Gamma 60, the MH 800 and the RW 400, are contrasted and an indication of the degree to which they fall short of a list of ideal characteristics is given.

1532
Analysis of Internal Computer Sorting by I. Flores; *J. Assoc. Comp. Mach.*, vol. 8, pp. 41–80; January, 1961.

Standard methods of internal sorting, such as sorting by insertion, counting, exchanging, selection, digital sorting, merging and various combinations and variations of the above, are compared and contrasted. The number of basic manipulations, such as comparisons, transfers, computer passes, and count modifications required for each method, is calculated. In this manner, the results permit the most advantageous method for a given application to be assessed.

6. FORMAL AND NATURAL LANGUAGES, INFORMATION RETRIEVAL, AND HUMANITIES

1533
Advanced Computer Applications by W. F. Bauer, D. L. Gerlough (Thompson Ramo Wooldridge), and J. W. Granholm; PROC. IRE, vol. 49, pp. 296–304; January, 1961.

Some new and interesting applications of stored-program digital computers are reviewed. Applications in literature and the arts, medicine, business and government, and "on-line" situations are briefly considered, and applications in communications, automobile traffic control, language translation, and industrial process control are considered in more detail. Applications referred to as "scientific" and "business" are omitted, as are computer technique except where these are important or peculiar to the applications.

1534
A System for Generating "Pronounceable" Names Using a Computer by A. L. Leiner and W. W. Youden (NBS); *J. Assoc. Comp. Mach.*, vol. 8, pp. 97–103; January, 1961.

A program for a computer-generated list of "pronounceable" names is described. The names all have four alphanumeric characters, which may be four letters, three letters and a numeral, or two letters and two numerals. A certain degree of error-detecting and error-correcting ability was built into the system.

1535
Information Retrieval and the Design of More Intelligent Machines by H. J. Gray, Jr. and E. J. Parker (University of Pennsylvania); *U. S. Govt. Res. Repts.*, vol. 35, pp. 323–324 (A); March 10, 1961. PB 153 597 (order from LC mi$4.50, ph$12.30).

A study of the operations involved in human cognitive process has resulted in an extension of the Newell-Simon Information Processing Languages (IPL) to a multiple-list data organization where each data item appears only once in an addressable memory, yet the item has in it several descriptors with control information that place the data item in several lists. Such a system has been shown to be well-suited to information retrieval problems. A specification of a parallel-access information bank, which may be more efficient than the multiple-list system, has been previously prepared. Central to the list machine and the parallel-access information-bank problem, is the problem of descriptor recognition. In this report, the problem is defined and progress towards its solution is stated.

1536
The Association Factor in Information Retrieval by H. E. Stiles (Dept. of Defense); *J. Assoc. Comp. Mach.*, vol. 8, pp. 271–279; April, 1961.

An all-computer document retrieval system which can find documents related to a request even though they may not be indexed by the exact terms of the request, and can present these documents in the order of their relevance to the request, is described. The key to this ability lies in the application of a statistical formula by which the computer calculates the degree of association between pairs of index terms. With proper manipulation of these associations (entirely within the machine) a vocabulary of synonyms, near synonyms and other words closely related to any given term or group of terms is derived. Such a vocabulary related to a group of request terms is believed to be a much more powerful tool for selecting documents from a collection than has been available heretofore. By noting the number of matching terms between this extended list of request terms and the terms used to index a document, and with due regard for their degree of association, documents are selected by the computer and arranged in the order of their relevance to the request.

A Magnetic Associative Memory—*see* 1496.

1537
A Card Format for Reference Files in Information Processing by M. Grems (IBM Corp.); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 90–98; February, 1961.

A card format suitable for a variety of reference files in information processing is proposed. An 80-column IBM card is divided into a flexible format reference field (cols. 1–67) and a rigid format identification field (cols. 68–80). The reference material includes an index, title, source, class, summary and cross reference for each entry. The identification consists of codes for descriptors, entry number, kind, major subject and source of the reference.

Storage and Retrieval of the Results of Clinical Research—*see* 1590.

7. BEHAVIORAL SCIENCE AND ARTIFICIAL INTELLIGENCE

1538
Recognition of Membership in Classes by G. S. Sebestyen (Melpar, Inc.); IRE TRANS. ON INFORMATION THEORY, vol. IT-7, pp. 44–50; January, 1961.

An approach to the general problem of recognition of membership in classes which are known only from a set of their examples is presented. A geometrical approach is taken where membership in classes is regarded measurable by metrics with which a set of points, representing different members of the same class, may be brought "close" to one another. For the case where classes are Gaussian processes, the method described and that of decision theory are found to agree. A practical application of the method to the automatically "learned" recognition of spoken numerals is described.

1539
Electronic Analog of the Human Recognition System by J. R. Singer (University of California); *J. Opt. Soc. Am.*, vol. 51, pp. 61–69; January, 1961.

A system for pattern recognition made up of electronic logic elements which has many of the characteristics of humans for recognizing patterns is described. In particular, the recognition is invariant for size and will tolerate a specified amount of tilt or figure rotation. The electronic components

or organs which make up the system consist of delay lines, logical elements such as AND circuits, and photoreceptors.

1540
Computer Identification of Vowel Types by J. D. Foulkes (Bell Telephone Labs.); *J. Acoust. Soc. Am.*, vol. 33, pp. 7–11; January, 1961.

In a classical study of the vowel sounds of English, G. E. Peterson and H. L. Barney collected a large body of experimental data which related perceived vowel quality to measurements of the first three formant frequencies and the voice pitch. It is difficult for a computer to use this raw data to interpret vowel quality because the vowel types have complicated boundaries in the coordinate system of the physical measurements. A coordinate transformation which simplifies these boundaries is described.

1541
Two Multivariate Statistical Computer Programs and their Application to the Vowel Recognition Problem by P. D. Welch and R. S. Wimpress (IBM Corp.); *J. Acoust. Soc. Am.*, vol. 33, pp. 426–434; April, 1961.

Two IBM 704 EDPM programs which were written to aid in the development of mechanical speech recognition devices are discussed. Both are based upon multivariate statistical techniques. The application of the two programs to the problems of vowel recognition using fundamental frequency and formant information is described.

1542
Preliminary Study of the Probabilistic Behavior of a Digital Network Majority Decision Element by S. Muroga (Rome Air Dev. Ctr.); *U. S. Govt. Res. Repts.*, vol. 35, p. 83 (A); January 13, 1961. PB 150 973 (order from LC mi$2.40, ph$3.30).

A majority decision element is an element in which a finite number of inputs are coupled with one output. The output value is one or zero, depending on the input values. An advantage of elements of this sort is that a single element can represent a fairly complex function. Consequently, the network for a given function can be represented with few elements. In addition, a highly reliable network can be constructed with unreliable elements. The probabilistic behavior of a network of elements characterized by a majority decision principle is discussed.

1543
Finite Automata, Pattern Recognition and Perceptrons by H. B. Keller (New York University); *J. Assoc. Comp. Mach.*, vol. 8, pp. 1–20; January, 1961.

A series of theorems formulating in terms of set theory the general problem posed by many automata is presented. The logic of a simple pattern-recognizing automaton is described, and the theory of a certain type of perceptron-like automata is developed. Some necessary conditions for discrimination by such automata are derived.

1544
Tables of *Q*-Functions for Two Perceptron Models by F. Rosenblatt (Cornell Aeronautical Lab.); *U. S. Govt. Res. Repts.*, vol. 35, p. 81 (A); January 13, 1961. PB 171 092 (order from OTS $2.75).

Four sets of tables for *Q*-functions are presented. These functions have been found to be essential in many quantitative analyses of perceptrons, and may also be of more general interest, since they include cumulative probability distributions for differences of binomial and Poisson distributed random variables.

1545
Contributions to Perceptron Theory by R. D. Joseph (Cornell Aeronautical Lab.); *U. S. Govt. Res. Repts.*, vol. 35, pp. 80–81 (A); January 13, 1961. PB 171 093 (order from OTS $2.50).

Perceptrons are a class of brain models which have been introduced in previous reports. There are several ways in which an analysis of a perceptron's ability to associate responses with stimuli may proceed. One approach is to place a distribution over the members of a subclass (in a specific manner) and then to analyze statistically the ability of a randomly chosen member of the subclass to perform a given task. A second approach considers the "optimum" characteristics for the components for perceptrons subject to a given set of constraints, and "optimum" operating procedures for given subclasses of perceptrons. Until this past year, almost all the analytic work done on perceptrons has followed the first approach. Quite recently, some powerful results for the second approach have been obtained for a subclass of perceptrons called simple perceptrons. The purpose of this report is to present the rigorous analyses of the first type which have been obtained, to display the techniques which have thus far proved useful, and to give some additional results of the second type.

A Simulator Study of a Two-Parameter Adaptive System—*see* 1621.

The Use of Threaded Lists in Constructing a Combined ALGOL and Machine-Like Assembly Processor—*see* 1530.

Information Retrieval and the Design of More Intelligent Machines—*see* 1535.

1546
A Generalized Technique for Symbol Manipulation and Numerical Calculation by D. T. Ross (M.I.T.); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 147–150; March, 1961.

A technique employing reversed use of index registers that provides a generalized method of handling problems requiring both computation and logical processing is described. The procedure leads to a type of generic structure called a plex, more powerful and general than lists or trees (which are included as special cases). The technique is better described by a flow diagram than by a sequential language.

1547
Proving Theorems by Pattern Recognition—II by H. Wang (Bell Telephone Labs.); *Bell Sys. Tech. J.*, vol. 40, pp. 1–41; January, 1961.

Theoretical questions concerning the possibilities of proving theorems by machines are considered from a viewpoint that emphasizes the underlying logic. A proof procedure for the predicate calculus that contains a few minor peculiar features is given. A fairly extensive discussion of the decision problem is given, including a partial solution of the $(x)(Ey)(z)$ satisfiability case, an alternative procedure for the $(x)(y)(Ez)$ case, and a rather detailed treatment of Skolem's case. In connection with the $(x)(Ey)(z)$ case, an amusing combinatorial problem is suggested. Some simple mathematical examples are also considered. (See also abstract 967.)

1548
Programming Intelligent Problem Solvers by W. R. Reitman (Carnegie Inst. Tech.); IRE Trans. on Human Factors in Electronics, vol. HFE-2, pp. 26–33; March, 1961.

Two research programs illustrating the evolution of heuristic programming systems are discussed. Applications of these techniques, with emphasis on methods and goals, in studies of problem solving are also considered.

1549
Man-Computer Cooperation in Decisions Requiring Common Sense by D. B. Yntema and W. S. Torgerson (Lincoln Lab., M.I.T.); IRE Trans. on Human Factors in Electronics, vol. HFE-2, pp. 20–26; March, 1961.

Three methods of enabling man to convey his decision rules to a machine are discussed. An attempt is made to foresee research problems.

8. MATHEMATICS

The Use of Index Calculus and Mersenne Primes for the Design of a High-Speed Multiplier—*see* 1475.

1550
New Factors of Mersenne Numbers by E. Karst (Brigham Young University); *Math. of Computation*, vol. 15, p. 51; January, 1961.

The Mersenne numbers 2^p-1, corresponding to prime exponents p in the interval $3000<p<3500$, have been tested for prime factors. The limit of the search for factors was $9p^2$ when no factor was previously known; otherwise, the limit was $3p^2$. The 19 new prime factors found by this search are tabulated.

1551
A Least Squares Surface Fitting Program by J. H. Caldwell (Royal Aircraft Estab.); *Computer J.*, vol. 3, pp. 266–269· January, 1961.

A program to fit a bivariate polynomial to a set of z values specified at points of a rectangular grid in the (x, y) plane is described. The method of orthogonal polynomials is used, and a detailed guide as to which terms are to be included is provided.

1552
Approximation of Curves by Line Segments by H. Stone (Shell Development Co.); *Math. of Computation*, vol. 15, pp. 40–47; January, 1961.

An analytical formulation of the problem of least-squares optimal approximation of a curve on an interval by broken line segments is given. In the quadratic case, it is shown that the break points in the approximating broken line should be equally spaced over the given interval. This allows for derivation

of a simple closed-form analytical solution. For the general case, a numerical method of solution involving a known FORTRAN IBM 704 computer program is presented. Applications of the solutions to the lead-susceptibility curve $f(x)=k_1+k_2e^{-cx}$ in the gasoline blending problem and to the more general curves $g(x)=f(x)+k_3x$ are noted. Tables of line parameters for two to four fitting lines and of maximal fitting error are given for the lead-susceptibility curve.

1553

A Note Concerning Orthogonal Polynomials by R. H. Bacon (Genl. Precision Lab.); *SIAM Rev.*, vol. 2, pp. 269–276; October, 1960.

Several sets of orthogonal polynomials suitable for the approximation of transcendental functions are constructed. Results are presented in five tables which list sets of polynomials of the following types: those containing all powers of x, those containing only the odd powers of x, those containing only the even powers of x, those without the constant term, and those with only even powers but without the constant term. In each case the set is orthogonal over an interval O to T. The tables also include recursion formulas for computing the higher degree polynomials.

Computer Generation of Optimized Subroutines—*see* 1526.

1554

Improved Formulas for Complete and Partial Summation of Certain Series by H. E. Salzer and G. M. Kimbro (Convair-Astronautics); *Math. of Computation*, vol. 15, pp. 23–39; January, 1961.

Formulas were previously given for summing a series to infinity (complete) or to a certain number of terms (partial) by considering the sum of the first j terms S_j, or a modification S_j' related to S_j, as a polynomial in $1/j$. The sum S_∞ or S_n was then found by m-point Lagrangian extrapolation from S_{j_0}, S_{j_0-1}, $\cdots$, S_{j_0-m+1} to $1/j=0$ or $1/j=1/n$. This paper gives more accurate m-point formulas for sums S_j which behave like even functions of $1/j$. Applications include the calculation of π, Euler's constant, Catalan's constant, a definite integral as the limit of a suitably chosen sequence, and the later zeros of the Bessel functions $J_r(x)$ from earlier zeros for suitable values of r.

1555

Comparison of Iterative Methods for the Calculation of *N*th Roots by J. F. Traub (Bell Telephone Labs.); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 143–145; March, 1961.

Three iterative methods (Newton's method, Bailey's third-order method, and a multiterm generalization of Newton's method) are compared for: 1) theoretical convergence, 2) actual machine running time for a variety of input data, and 3) storage space in a computer. It is concluded that the multiterm method converges the most rapidly, and Bailey's method is the fastest, but Newton's method has a slight storage advantage over Bailey's, which in turn requires far less storage than the multiterm method.

1556

A Note on Systems of Linear Equations by D. L. Elliott (U. S. Naval Ordnance Test Station); *SIAM Rev.*, vol. 3, pp. 66–69; January, 1961.

A generalization of Semarne's method of solving systems of equations is presented. Given a system of m linear equations in n unknowns (1) $Ax=c$, three cases are distinguished: (I) no finite vector satisfies (1), (II) a unique vector satisfies (1), (III) an infinity of vectors satisfies (1) such that their endpoints lie on some line, plane, or higher-dimensional linear manifold. Case (II) for $m=n$ was considered by Semarne. The method is modified to handle any system (1) whether or not A has an inverse, giving a solution if one exists and indicating if a solution does not exist.

1557

A Theorem on Determinants by E. Gott (University of Hawaii); *SIAM Rev.*, vol. 2, pp. 288–291; October, 1960.

Let Δ be a determinant of order k with general element $\sigma_{mn}=\alpha_{mn}+i\beta_{mn}$. Let Ω be the determinant of order $2k$ whose odd-numbered rows are given by $(\alpha_{j1}, \beta_{j1}, \alpha_{j2}, \beta_{j2}, \cdots, \alpha_{jk}. \beta_{jk})$ where $j=1, 2, \cdots, k$ in turn, and the even-numbered rows are given by $(-\beta_{j1}, \alpha_{j1}, -\beta_{j2}, \alpha_{j2}, \cdots, -\beta_{jk}, \alpha_{jk})$ where $j=1, 2, \cdots, k$ in turn. It is then proved that $\Omega\equiv\Delta\bar{\Delta}$ where $\bar{\Delta}$ is the complex conjugate of Δ.

1558

Two Theorem Tables of Matrix Algebra by G. C. Best; *Math. of Computation*, vol. 15, pp. 19–22; January, 1961.

By suitable modification of the statements of matrix algebra theorems, a large number of these theorems can be presented in tabular form. For example, defining an "inverse-forming" operator O^{-1} by $O^{-1}A=A^{-1}$ (A nonsingular) and a transposing operator O^T by $O^T\ A=A^T$, the theorem $(A^T)^{-1}=(A^{-1})^T$ can be written as $O^TO^{-1}A=O^{-1}O^TA$ and thus presented in an "operator vs operator" table by writing k (denoting commutativity) in the entry corresponding to row O^T and column O^{-1}. A table (Operators vs Properties) lists theorems corresponding to 18 operators and 22 properties, and a second table (Operators vs Operators) lists theorems corresponding to 22 operators.

1559

An Iterative Method of Numerical Differentiation by D. B. Hunter (GE Co. Ltd.); *Computer J.*, vol. 3, pp. 270–271; January, 1961.

A method for estimating the derivative of a function defined by a set of values is described. The method is similar to those of Aiken and Neville for iterative interpolation, and like them can be used with either equally or unequally spaced ordinates. The derivative is obtained for any value of the argument, not just for the table values. A disadvantage is that the function y as well as its derivative y' must be estimated.

1560

A Bibliography on Approximate Integration by A. H. Stroud (University of Wisconsin); *Math. of Computation*, vol. 15, pp. 52–80; January, 1961.

A bibliography on various subjects related to approximate integration, including quadrature (or numerical integration) formulas and finite-difference approximations, is presented. Papers dealing with error theory are also covered, but applications of integration methods to the solution of differential equations, graphical methods, and tables of rational function approximations for functions expressed as integrals are omitted.

1561

Numerical Integration Using Sums of Exponential Functions by F. C. Ledsham; *Math. of Computation*, vol. 15, pp. 48–51; January, 1961.

An approximation to the integral

$$\int_{x_0+mh}^{x_0+kh} q(x)dx$$

is made by replacing $q(x)$ by a sum of exponential functions

$$f(x) = \sum_{i=0}^{n} \alpha_i \exp(\alpha_i x),$$

leading to an expression of the form

$$\int_{x_0+mh}^{x_0+kh} f(x)dx = \sum_{r=0}^{n} B(k, m, n; h, r)q_r$$

where $q_r=q(x_0+rh)$.

Analytical formulas from which the required coefficients $B(k, m, n; h, r)$ can be directly obtained are presented. Results are applied to examples previously given by Greenwood and by Brock and Murray.

1562

New Tables of Howland's and Related Integrals by C. W. Nelson (IBM Corp.); *Math. of Computation*, vol. 15, pp. 12–18; January, 1961.

Howland's integrals are defined by

$$\left.\begin{matrix} I_k \\ I_k^* \end{matrix}\right\} = \frac{1}{2(k!)}\int_0^\infty \frac{w^k dw}{\sinh w \pm w} \left\{\begin{matrix} k \geq 1 \\ k \geq 3 \end{matrix}\right.$$

$$\left.\begin{matrix} II_k \\ II_k^* \end{matrix}\right\} = \frac{1}{2(k!)}\int_0^\infty \frac{w^k e^{-w} dw}{\sinh w \pm w} \left\{\begin{matrix} k \geq 1 \\ k \geq 3 \end{matrix}\right.$$

These integrals were previously tabulated by Ling and Nelson to 6D. In this paper their values are given to 9D for $k=1, 2, \cdots, 36$. Values of I_{2k}, I_{2k}^*, and $\left[{2 \atop n}\right]$ are given to 18D, for $k=1, 2, \cdots, 33$, and $n=1, 2, \cdots, 40$, where

$$\left[{2 \atop n}\right] = \frac{1}{2(2!)}\int_0^\infty \frac{x^{n+1}}{\sinh^n x}dx.$$

Application is made to the evaluation of related integrals, in particular to

$$\int_0^\infty \frac{xJ_1(\rho x)dx}{\sinh x + x}.$$

1563

Convergence Properties of a Gaussian Quadrature Formula by W. Barrett (University of Leeds); *Computer J.*, vol. 3, pp. 272–277; January, 1961.

The remainder of any Gaussian quadrature formula is expressed as a contour integral, which is then used to obtain error estimates in certain cases. Numerical examples comparing the actual and estimated errors are given. A method of constructing

more general quadrature formulas is presented, and two examples of such formulas are constructed.

1564
Recursive Computation of Certain Integrals by W. Gautschi (Oak Ridge Natl. Lab.); *J. Assoc. Comp. Mach.*, vol. 8, pp. 21–40; January, 1961.

When problems such as the evaluation of certain integrals are solved by means of a linear recurrence relation, it sometimes happens that considerable significance is lost by subtractive cancellation. The situation may often be improved by backward rather than forward recurrence. Criteria for determining the error buildup in each case and hence for choosing the direction of recursion are provided.

1565
The Philosophy and Applications of Transform Theory by M. S. Klamkin (Avco) and D. J. Newman (Yeshiva University); *SIAM Rev.*, vol. 3, pp. 10–36; January, 1961.

The philosophy of transform theory is summarized in three basic steps: 1) transformation of an original difficult problem to an easy problem, 2) solution of the easy problem, and 3) inversion to obtain the solution to the original problem. The following examples are given to illustrate the underlying philosophy: determination of the maximum (and minimum) area bounded by three parallel, mutually tangent, congruent ellipses; Hilbert's problem on the impossibility of constructing the center of a circle with only a straightedge; applications of geometrical inversion to a Steiner chain of circles, to the problem of Appolonius, and to the problem of constructing the center of a given circle with compass alone; solution of a diophantine equation; applications of generating functions; principles of duality in differential equations; Riemann mapping; Laplace transform; and generalized integral transforms.

1566
Conformal Transformations by Analogue Computer by E. J. Joss and D. S. Ross (Royal College Sci. and Tech.); *Brit. J. Appl. Phys.*, vol. 12, pp. 178–179; April, 1961.

The use of an analog computer to generate certain conformal transformations of the circle, which are required in the analysis of stress patterns in the neighborhood of holes of various shapes, is described. The relationship between the form of transformation and the shape produced is also discussed.

1567
Special Block Iterations with Applications to Laplace and Biharmonic Difference Equations by H. B. Keller (New York University); *SIAM Rev.*, vol. 2, pp. 277–287; October, 1960.

Some iterative methods for solving systems of linear equations $M\phi=b$ (M positive definite) are investigated. The matrix M is of the form $M=I-H-V$ where $HV=VH$, $H^T=H$, and $V^T=V$. Systems of this kind arise in the finite-difference solution of elliptic and parabolic partial differential equations. The iteration is of the form $N\phi^{(n+1)}=P\phi^{(n)}+b$, $(n=0, 1, 2, \cdots)$, where $N-P=M$, $|N|\neq 0$, and the splitting of M is given by $N(\alpha, \beta, \gamma)=\alpha I-\beta H-\gamma V$, $P(\alpha, \beta, \gamma)=(\alpha-1)I-(\beta-1)H-(\gamma-1)V$. It is shown that convergent schemes are easily found, and some results on obtaining a best scheme are presented. Best schemes are obtained for some Laplace and biharmonic difference equations.

1568
Families of Sturm-Liouville Systems by E. L. Dunn (U. S. Naval Ordnance Test Station) and F. M. Stein (Colorado State University); *SIAM Rev.*, vol. 3, pp. 54–65; January, 1961.

A Sturm-Liouville system consists of a differential equation

$$(1)\quad py''+sy'+(q+\lambda_i r)y=0$$

and linear homogeneous bounary conditions

$$(1')\quad \begin{aligned} Ay(a)+By'(a)&=0,\\ Cy(b)+Dy'(b)&=0,\end{aligned}$$

where p, q, r, s are real analytic functions of x on an interval $[a, b]$, $p(x)>0$ in this interval, λ_i is a parameter, and A, B, C, D are constants. A family of Sturm-Liouville systems is defined as a collection of such systems consisting of (1) and (1′) and all systems which can be obtained by repeated differentiation and integration of (1) and application of boundary conditions of form (1′). Conditions on the coefficients of (1) are given for the generation of a family and theorems analogous to those for a general Sturm-Liouville system are presented for families. It is shown that the sets of eigenvalues for nontrivial eigenfunctions are identical for all members of a family of Sturm-Liouville differential equations. Applications to the Hermite, Jacobi, and Laguerre equations are given.

1569
On Finding Minimum Routes in a Network with Turn Penalties by T. Caldwell (University of Arizona); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 107–108; February, 1961.

The work of Moore and Pavley is extended to include networks with turn penalties. A turn penalty t_{ijk} is a positive number associated with a given pair of links (i, j) and (j, k) of a linear graph. Best routes in networks with turn penalties do not necessarily coincide with best routes in the same network without turn penalties. A method of reducing a network with turn penalties to one without turn penalties is presented.

1570
On the Exceptional Case in the Characterization of the Arcs of Complete Graph by A. J. Hoffman (GE Co.); *IBM J. Res. and Dev.*, vol. 4, pp. 487–496; November, 1960.

A simple set of properties characterizing the relationship of adjacency among the arcs of the complete graph or order n, when $n=8$, is reviewed. A method for enumerating all the anomalous cases for $n=8$ is described.

1571
Applications of Graphs and Boolean Matrices to Computer Programming by R. B. Marimont (Natl. Inst. of Health); *SIAM Rev.*, vol. 2, pp. 259–268; October, 1960.

The elementary properties of directed graphs and Boolean matrices are reviewed and the relationship of their uses in other fields to computer programming is shown. The basic definitions of graphs are introduced, and the use of the reachability matrix of a program for the detection of blind alley errors is discussed. Some unsolved problems associated with the mechanization of computer programming are also considered.

1572
Some Combinatorial Lemmas in Topology by H. W. Kuhn (IBM Corp); *IBM J. Res. and Dev.*, vol. 4, pp. 518–524; November, 1960.

From a simple proof for the Sperner Lemma in two dimensions, *i.e.*, in any subdivision of a properly labelled triangle, at least one of the subdividing triangles must be properly labelled; the well-known Brouwer Fixed Point Theorem is shown to follow naturally. Results analogous to the Sperner Lemma are proved for the n-cube. The relation of these results to previous theorems due to Ky Fan and Tucker is discussed.

9. PROBABILITY, INFORMATION THEORY, AND COMMUNICATION SYSTEMS

Predicting Distributions of Staff—*see* 1465.

Time-Optimal Control of Higher-Order Systems—*see* 1622.

1573
Optimum Prediction with a Mean Weighted Square Error Criterion by C. C. Glover (Johns Hopkins University); IRE TRANS. ON AUTOMATIC CONTROL, vol. AC-6, pp. 43–48; February, 1961.

The linear prediction theory is examined using a mean-weighted square-error criterion. A specific nondeterministic weighting function is used. The problem is reduced to that of solving integral equations which are written in terms of correlation functions which can be calculated by averaging over the ensemble. A complete solution for the problem using Gaussian statistics with no correlation between noise and true signal is given.

1574
Statistical Programs at the University of North Carolina by N. Bush (University of North Carolina); *Commun. Assoc. Comp. Mach.*, vol. 4, pp. 108–109; February, 1961.

The main features of three statistical programs—1) a General Contingency Table Analysis for Questionnaire Data, 2) Analysis of Variance, and 3) Multiple Regression and Correlation—are described.

1575
Multiple Regression Analyses Program for the IBM Type 650 by C. E. Wright (Washington University); *U. S. Govt. Res. Repts.*, vol. 35, p. 191 (A); February 10, 1961. PB 148 732 (order from LC mi $1.80, ph $1.80).

The program computes β matrices and multiple correlations according to the following matrix equations: (1) $r_{mm}^{-1}T_{mm}=\beta_{mm}$ and (2) $\beta_{nm}T_{mn}=R_{nn}$, where r_{mm} represents the intercorrelation matrix of predictor variables, T_{mn} a matrix of the correlations of the m predictor variables with the n criterion variables, β_{mn} the m by n matrix of regression weights, and R_{nn} a matrix whose diagonal

elements are the squares of the multiple correlations of the predictor variables with the criterion variables. The closeness of the inverse involved can optionally be checked by computing the identity matrix. The program is restricted to m, the number of predictor variables, equal to or less than 38 and n, the number of criterion variables, equal to or less than 50 with the further restriction that the product mn be less than 1600. The input and output with the exception of the inverse are in fixed point form, but the computations are performed in standard 8-digit floating point. Provisions were made so that the outputs of the symmetric and nonsymmetric correlation matrix programs (AD-204 459) may serve as input.

1576
A Comparison of Some Methods of Calculating Covariance Functions on an Electronic Computer by I. J. Good (Admiralty Res. Lab.); *Computer J.*, vol. 3, pp. 262–265; January, 1961.

Three methods of calculating covariance functions of two sequences of numbers are compared. The number of digits in each number is assumed small in comparison with the machine word length. The methods describe alternative means of packing several numbers to a computer word, in order to take maximum advantage of the multiply instruction.

1577
Random Sampling from the Normal Distribution by J. C. Butcher (University of Sydney); *Computer J.*, vol. 3, pp. 251–253; January, 1961.

Two methods, one simple and one fast, for obtaining a sampling from a normal distribution are discussed. Assuming convenient means for generating random numbers and logarithms are available, a simple means of obtaining a sampling of the logarithmic distribution is described. Methods for checking whether individual samplings from this latter distribution are also acceptable as samplings of the normal distribution are considered.

The Backboard Wiring Problem: A Placement Algorithm—*see* 1482.

Generalized Simulation of Post Office Systems—*see* 1628.

The Theoretical Channel Capacity of a Single Neuron as Determined by Various Coding Systems—*see* 1592.

1578
On Coded Phase-Coherent Communications by A. J. Viterbi (California Inst. Tech.); IRE Trans. on Space Electronics and Telemetry, vol. SET-7, pp. 3–14; March, 1961.

The result of encoding independent binary digits into sets of binary code words which are transmitted over a channel perturbed by additive white Gaussian noise and detected by correlating them with their stored or locally generated replicas at the receiver is discussed. The word error probabilities and bit error probabilities for low cross-correlation codes are determined as a function of the ratio

$$\frac{\text{(received signal energy)/bit}}{\text{(noise power)/(unit bandwidth)}}.$$

The received information rate and the potential channel capacity are also computed. It is shown that in the limit as the code word length and the bandwidth approach infinity, the received information rate approaches the channel capacity for only one value of the above ratio.

1579
On the Construction of Minimally Redundant Reliable System Designs by D. K. Ray-Chaudhuri (University of North Carolina); *Bell Sys. Tech. J.*, vol. 40, pp. 595–611; March, 1961.

A general mathematical theory for generating minimally redundant error-correcting codes for synchronous digital systems such that output is free of error when there is a fault in at most one block of the system is presented. The problem of constructing minimally redundant reliable systems of the above type is completely solved. A detailed example of the application of the theory is supplied.

1580
Minimum-Redundancy Coding for the Discrete Noiseless Channel by R. M. Karp (IBM Corp.); IRE Trans. on Information Theory, vol. IT-7, pp. 27–38; January, 1961.

Albegraic method for constructing minimum-redundancy prefix codes for the general discrete noiseless channel without constraints is described. The costs of code letters need not be equal, and the symbols encoded are not assumed to be equally probable. First, structure functions are defined, in terms of which necessary and sufficient conditions for the existence of prefix codes may be stated. From these conditions, linear inequalities which may be used to characterize prefix codes are derived. Gomory's integer programming algorithm is then used to construct optimum codes subject to these inequalities; computational experience is presented to demonstrate the practicability of the method. Finally, some additional coding problems are discussed and a problem of classification is treated.

1581
Linear-Recurrent Binary Error-Correcting Codes for Memoryless Channels by W. L. Kilmer (Montana State College); IRE Trans. on Information Theory, vol. IT-7, pp. 7–13; January, 1961.

The analysis of recurrent-type, parity-check, error-correcting codes for memoryless, binary symmetric channels is considered. These codes are defined to consist of message sequences augmented by insertions of r successive parity digits for every b successive message digits. An analysis framework is established for the codes which consists mainly of a parity-check matrix $[M]$ and a message difference vector $[N]$. Within this framework, a decoding scheme is developed which renders the codes capable of correcting any set of $\leq e$ errors in m/b successive $(b+r)$ digit blocks of coded message sequence, where e is maximized over all parity-check codes having the same redundancy ratios and maximal lengths of dependence among their digits. An example of a linear-recurrent code which has a lower probability of error than the best comparable block code is given and several outstanding problems are discussed.

1582
Single Error-Correcting Codes for Nonbinary Balanced Channels by C. W. Helstrom (Westinghouse Res. Labs.); IRE Trans. on Information Theory, vol. IT-7, pp. 2–7; January, 1961.

Close-packed, single error-correcting codes, the letters of which are N-tuples of M-ary digits, where M is the power of a prime, are studied. The length N of the letters must be given by $N=(M^k-1)/(M-1)$, where k is an integer. A balanced communication channel, for which all errors in a transmitted digit are equally likely, is defined and a physical model is given. The probability of correct reception of the code letters and the rate with which they transmit information in a balanced channel is calculated. This involves deriving formulas for the numbers of code letters having various numbers of 0's. Numerical results are given for a quaternary code and are extended to the case where the quaternary channel has a null zone, so that erasures as well as errors may occur. For the type of signals and noise assumed, the balanced channel without a null zone is found to yield the better performance.

1583
Error-Correcting Codes for Multiple-Level Transmission by J. MacWilliams (Bell Telephone Labs.); *Bell Sys. Tech. J.*, vol. 40, pp. 281–308; January, 1961.

A q-level alphabet is defined as a row vector space over a finite field with q elements. The letters of the alphabet are the rows of the vector space, each consisting of n symbols from the ground field. The weight of a letter is the number of nonzero symbols it contains. The minimum weight of the letters of the alphabet, excluding zero, is denoted by d. A relationship is established between the alphabet and a set of points S in a finite projective space. There is a many-one correspondence between the letters of the alphabet and the hyperplanes of the space. The weight of a letter is simply related to the incidence of the set S with the corresponding hyperplane. Two sets of points in a finite projective space are called equivalent if they are related by a collineation of the space. Two alphabets are called equivalent if there exists between them, as vector spaces, a weight-preserving semi-isomorphism. It is shown that these definitions mean the same thing and reduce to the usual definition when $q=2$. An inequality is established between the dimension of the alphabet and the parameters d, q, n. This gives a lower bound for n in terms of the other parameters. It is shown that this bound cannot be achieved by alphabets with repeated columns. A method for con-

structing a class of alphabets which attain this bound is given. It is shown that for the case $q=2$, these are the only alphabets (in the sense of equivalence) for which the bound is attained.

Advanced Computer Applications—*see* 1533.

1584
Use of Digital Computer to Determine Broadcast Station Nighttime Interference by I. Akerman (Dept. of Transport, Ont.); IRE TRANS. ON BROADCASTING, vol. BC-7, pp. 5–11; March, 1961.

A computer program to determine nighttime sky-wave signal interference to groundwave service is described. The formulas and sequences used by the program are discussed. A sample calculation of the program is given. The advantages of such a program, to the broadcast design engineer, are accuracy, reliability, speed and convenience.

1585
An Improved Decision Technique for Frequency-Shift Communications Systems by E. Thomas (Page Communications Engrs., Inc.); PROC. IRE, vol. 48, pp. 1998–2003; December, 1960.

A new circuit technique computer called a decision threshold which enables a frequency-shift-keyed receiver system to use information in mark and space channels independently, resulting in an improvement in circuit quality where fading exists between the mark and space frequencies, is described. Typical FSK demodulators, with a fixed decision level set halfway between the long-term average amplitudes of the receiver mark and space frequencies, are shown to result in a high error liability when deep fades occur on either the mark or space frequency. It is further shown that such errors are unnecessary since the complete message is available on either frequency. The device discussed has been designed to make use of the normally deleterious effects of frequency selective fading, to provide up to 30-fold reduction in teleprinter error rates over that which is theoretically possible when flat fading is assumed. Additional orders of diversity are shown to result where low correlation exists between the mark and space frequencies.

10. SCIENCE, ENGINEERING AND MEDICINE

1586
The Use of Automatic Digital Computation Machine in Design of Miniature Pulse Transformers and Power Transformers by D. Wildfeuer (American Bosch Arma Corp.); *Proc. Natl. Electronics Conf.*, vol. 16, pp. 631–651; 1960.

The application of a general-purpose digital computer to the design of miniature pulse transformers and power transformers is described. For each type of component, a flow chart which describes the computation flow is presented. For each of the categories, the machine prints out computed electrical and magnetic characteristics in association with coil details and core structure. Sample problems of machine designed pulse transformers and a power transformer are presented.

1587
Optimum-Allocation of Discharge to Units in a Hydro-Electric Generating Station by B. Bernholz (Hydro-Electric Power Comm., Ont.); *SIAM Rev.*, vol. 2, pp. 247–258; October, 1960.

The output of a generating station, corresponding to a given station discharge, depends on the division of the discharge among individual generating units in the station. The problem under consideration is to determine the division of discharge for maximum output. A solution which is sufficiently general to cover all practical cases is presented, and a simple computational procedure is included. Advantage is taken of the fact that individual units in a station can usually be divided into groups such that the units in any one group are identical. For each such group, the allocation of discharge to individual units is determined. Total station discharge is then divided among the groups so as to produce maximum station output. A numerical example is given for the case of two groups, each with three identical units.

1588
On Samulon's Formula for Frequency Response from Step Response by B. W. Lindgren (University of Minnesota); *SIAM Rev.*, vol. 1, pp. 47–49; January, 1959.

A simpler and less suspect derivation of Samulon's formula which avoids the manipulation of divergent series is given. The formula is embodied in the theorem: if $K(t)$ is of bounded variation, if $Y(i\omega)$ is its Fourier-Stieltjes transform, and if $Y(i\omega)$ vanishes identically outside $-W<\omega<W$, $(W=2\pi f_c)$, then $Y(i\omega)$ is respresentable, at each point in the interval where it has a left- and a right-hand derivative, in the form

$$Y(i\omega) = [\exp(-i\omega/4f_c)(\omega/4f_c)[\sin\omega/4f_c]^{-1} \cdot \sum_{-\infty}^{\infty} B_n \exp(-in\omega/2f_c)$$

where

$$B_n = K(n + 1/2f_c) - K(n/2f_c).$$

The effect of the presence of frequencies above the assumed cutoff frequency is also described.

A Semiconductor Binary Coordinate Converter—*see* 1484.

Automatic Processing System for Acoustical Data—*see* 1626.

1589
On the Design of Optical Systems with an Aspheric Surface by K. Miyamoto (University of Rochester); *J. Opt. Soc. Am.*, vol. 51, pp. 21–22; January, 1961.

A method of designing a correcting aspherical surface which is situated in an arbitrary position within an optical system is described. The method requires the solution of two first-order differential equations and seems more suitable for use with an electronic computer than methods previously proposed by others. It can be considered as a differential form of a method previously proposed by E. Wolf.

1590
Storage and Retrieval of the Results of Clinical Research by M. Eden (M.I.T. and Natl. Heart Inst.); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 265–268; October, 1960.

The application of structural linguistics to machine indexing and abstracting for the retrieval of data pertinent to specific clinical problems is described.

1591
Major Problems in the Use of Computing Machines by R. Taylor (IBM Res. Ctr.); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 253–254; October, 1960.

The questions of machine-recorded medical histories and diagnoses, the machine-handling of laboratory data, and graphical and pictorial information as well as communication between doctor and machine are considered.

1592
The Theoretical Channel Capacity of a Single Neuron as Determined by Various Coding Systems by A. Rapoport and W. J. Horvath (University of Michigan); *Information and Control*, vol. 3, pp. 335–350; December, 1960.

The information channel capacity of a model neuron with a fixed refractory period δ is calculated for optimum continuous time interval coding. Two types of noise perturbations are considered, a Gaussian probability distribution and a rectangular distribution of the time of occurrence of the response to a stimulus. Laboratory measurements indicate that a Gaussian distribution with a standard deviation of latency, σ, of about 5 μsec gives the best fit to an actual nerve fiber. This results in a maximum information transmission of slightly over 4000 bits per second. The results are compared with the discrete theory of MacKay and McCulloch.

1593
Short-Term Memory in Vision by E. Averbach and A. S. Coriell (Bell Telephone Labs.); *Bell Sys. Tech. J.*, vol. 40, pp. 309–328; January, 1961.

Experiments that demonstrate some of the functional properties of short-term storage in the visual system, its decay, readout and erasure are described. Results indicate that the visual process involves a buffer storage which includes an erasure mechanism that is local in character and tends to erase stored information when new information is put in. Storage time appears to be of the order of one-quarter second; storage capacity is more difficult to assess.

1594
Computers and Clinical Psychiatry by P. G. S. Beckett (Lafayette Clinic); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 248–250; October, 1960.

A method of recording psychiatric interviews in a form suitable for high-speed data processing is presented. Its advantages and disadvantages are discussed.

1595
Digital Computers and Medical Logic by R. Ebald and R. Lane (RCA); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 283–288; October, 1960.

The logical structure of a computer program designed to analyze and determine significant relationships between sets of symptoms—those drawn from case histories and those from a "classical" set—in a group of hematological diseases is discussed.

1596
Correlation of Data with a Digital Computer in the Differential Diagnosis of Hematological Diseases by M. Lipkin (Bellevue Hosp.); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 243–246; October, 1960.

The application of a digital computer to a comparison between hospital case data and data characteristic of hematologic diseases is described.

1597
Diagnosis of Arterial Disease with Analog Computers by R. W. Stacy (Ohio State University); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 269–273; October, 1960.

Using an analog computer to simulate the behavior of the arterial system, it is possible to compute a peripheral pulse wave which closely approximates the peripheral pulse wave recorded from a patient, when the central pulse wave recorded from that patient is used as a forcing function. A second-order differential equation is used, describing primarily the vibratory behavior of the system. When behavior duplication is verified, the equation parameters may be analyzed to provide data on arterial elasticity and blood viscosity and mass.

1598
Diagnostic Video Data Processing by L. B. Lusted (University of Rochester); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 293–294; October, 1960.

Preliminary studies of a spatially-oriented computer that can "read" chest X-ray photofluorograms and separate the normal chest films from the abnormal are reported.

1599
Use of a Digital Computer in the Analysis of Intestinal Motility Records by J. T. Farrar (Massachusetts Memorial Hosp.); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 259–263; October, 1960.

A method for the rapid quantitative analysis of complex phasic patterns by converting these waveforms into digital form is described. Generation of the autocorrelation function and the power density spectrum of these records has permitted a numerical, quantitative expression of certain information contained in these curves.

1600
Some Problems and Approaches to Automation of Medical Diagnosis by P. A. Smith (System Development Corp.); *Behavioral Sci.*, vol. 6, pp. 88–91; January, 1961.

Problems encountered in medical diagnosis by computers are discussed. It is pointed out that completely objective physiological data may not always be obtained from laboratory tests and instrumented data-gathering techniques. This problem is being overcome by the recent development of continuously transmitting telemetering devices for use in the "man in space" and "animal in space" programs. Nonphysiological data can be evaluated by judges or by means of check lists. Some other medical applications of computers are also briefly discussed. These include the use of computers in hospital administration and in the development of new drugs and pharmaceuticals.

1601
Some Reflections on Medical Diagnosis by Electronic Data Processing Machines by G. R. Meneely (Vanderbilt University); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 309–313; October, 1960.

The feasibility of computer diagnosis is discussed with emphasis on coordinating engineering and medical effort. It is concluded that government services, industry, and medical centers would be primary sites.

1602
Computer Programming of Diagnostic Tests by L. B. Lusted (University of Rochester); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 255–258; October, 1960.

Use of electronic data processing to help determine the minimum additional medical tests needed in a specific case is discussed.

1603
Diagnostic Aspects of Computer Applications in Medical Research at the University of Pennsylvania by M. L. Rockoff (University of Pennsylvania); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 250–252; October, 1960.

The diagnostic implications of medical research utilizing a digital computer according to the type of computer technique employed are discussed. Illustrative examples include the solution of differential equations in anesthesiology, the use of Fourier analysis in ballistocardiology, and the use of multiple regression analysis in neoplastic chemotherapy.

1604
Digitation of Clinical and Research Data in Serial Evaluation of Disease Processes by W. A. Spencer (Baylor University) and C. Vallbona (Texas Inst. for Rehabilitation and Res.); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 296–308; October, 1960.

The information flow between patient and physician that is required to develop extensive application of EDP for clinical use as well as for research in chronic disease is discussed. The advantages, disadvantages, and technical problems of EDP are considered.

1605
IBM Type 704 Medical Diagnosis Program by T. Tanimoto (IBM Res. Ctr.); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 280–283; October, 1960.

A method of computer diagnosis based on pairing symptoms with medical cases is described. By interpreting pairwise similarity as probabilities, it is possible to predict how likely it is for a case with given symptoms or tests to exist on the basis of the classification the computer produces.

1606
Computers and Psychophysiology in Medical Diagnosis by A. F. Ax (Lafayette Clinic); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 263–264; October, 1960.

Apparatus designed to sample, digitize, and magnetically record for computer input up to 29 physiological variables as recorded on the intact human is described.

1607
Doctor-Machine Symbiosis by J. J. Baruch (Bolt, Beranek and Newman); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 290–293; October, 1960.

The facilitation of communication between doctor and machine is considered. To study methods of communication, to match "communication rates," and to establish those parts of the diagnostic problem for which doctor and machine are best suited are the most important part of machine-aided diagnosis.

1608
An Analog Approach to Computer Diagnosis by L. J. Brannick (Princeton Sci. Assoc.); IRE TRANS. ON MEDICAL ELECTRONICS, vol. ME-7, pp. 247–248; October, 1960.

A method of diagnostic computation by defining each disease in terms of its symptoms and their relative significance in the characteristics of that disease is described.

11. ANALOG AND HYBRID COMPUTERS

1609
Testing Continuous Computers by R. C. Mikulich (University of Chicago); *U. S. Govt. Res. Repts.*, vol. 35, p. 81 (A); January 13, 1961. PB 171 080 (order from OTS $3.50).

A mathematical theory of testing continuous computers is given and the differences between component and system testing are delineated. An appropriate component test program, with interpretation and analysis, is proposed. The use and application of system testing is then discussed and criteria for the selection of test problems, their running, and subsequent analysis are established. One chapter is devoted to the operational aspects of a computer, including quantitative measures of repeatability and reliability and qualitative discussions of recordability, adaptability, and convenience.

1610
Test of a Model Dynamic System Synthesizer by R. C. Mikulich (University of Chicago); *U. S. Govt. Res. Repts.*, vol. 35, pp. 189–190 (A); February 10, 1961. PB 171 144 (order from OTS $3.00).

A mathematical (as opposed to engineering) test program conducted in conjunction with a program to design and develop performance tests for continuous computers and involving unique tests, for an experimental 18-amplifier analog computer is described. A comprehensive analysis of the components, a full account of the system tests, a discussion of individual system and results,

and a presentation of operational aspects are included. The theoretical basis for this program can be found in PB 171 080.

Discrete Analog-Computer Compensation of Sampled-Data Control Systems—*see* 1624.

1611
An Analog Correlation Computer by C. E. McCullough (University of Texas); *U. S. Govt. Res. Repts.*, vol. 35, p. 81 (A); January 13, 1961. PB 150 649 (order from LC mi $3.60, ph $9.30).

A correlation computer designed to be compatible with equipment presently being used for data handling is described. The computer obtains its input from magnetic tape, and since the time functions to be analyzed are recorded directly on magnetic tape, correlation analyses may be performed using a minimum of manual or automatic transcription processes. Time functions from a few seconds to several hours in length may be analyzed. A discussion of the theory involved in correlation analysis, a description of the operation of the computer, and several examples of autocorrelation and cross-correlation analyses which have been performed with this computer are also included.

1612
An Electronic Analogue Computer for a Coal Transportation Problem by E. G. Anderson (IBM U.K. Ltd.); *Proc. IEE*, vol. 108B, pp. 43–47; January, 1961.

The transportation problem and linear programming are defined and an electronic analog of the problem and its optimum solution is described. A computer with a 50×20 matrix has been built using current and opposing EMF to simulate coal flow and route costs, respectively. The circuits and the operating technique are described. A typical problem and its solution are given, and it is concluded that a larger computer is feasible.

1613
Transistorized Electronic Analog Multiplier by S. Deb and J. K. Sen (Calcutta University); *Rev. Sci. Instr.*, vol. 32, pp. 189–192; February, 1961.

The exponential current-voltage characteristic of the input of a grounded base junction transistor is utilized to construct an analog multiplier. Four-quadrant operation is obtained by using two channels—one with *p-n-p* and the other with *n-p-n* transistors. Design considerations of the various components of the multiplier are discussed. It is shown that the performance of the multiplier, particularly with respect to bandwidth, compares favorably with that of the other types described in the literature.

1614
A Function Generator Using Cold-Cathode Selector Tubes by R. M. Duffy (Natl. Transformers) and C. P. Gilbert (University of New South Wales); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-10, pp. 71–77; March, 1961.

A method of generating voltages which are arbitrary functions of time by using a chain of cold-cathode selector tubes as a single-pole, multiposition switch is suggested. Accuracies of ±1 per cent can be achieved with relatively simple adjustment. A generator using this method is described in detail, and typical output curves are shown. The generator is extremely versatile, not only in the range of functions which can be produced, but also in its ability to: 1) operate over a wide range of speeds, 2) change instantaneously from one speed to another, and 3) generate two separate functions, one displaced with respect to the other by a variable, preset delay.

1615
A Versatile Forcing Function Generator by J. Morrison (English Electric Aviation); *Electronic Engrg.*, vol. 33, pp. 155–159; March, 1961.

A forcing function generator, designed for analog computer use, which will provide sine, square, triangular, ramp, step or impulse functions to an accuracy of ¼ per cent full-scale in frequency and amplitude is described. Frequency coverage is from 0.001 cps to 100 cps, and the device can be centrally controlled from the computer, or independently.

1616
Memory Cells for Analog Computers by T. A. Bickart and R. P. Dooley (Johns Hopkins University); *Electronics*, vol. 33, pp. 71–73; December 9, 1960.

An analog computer memory cell that holds both positive and negative signal levels within 10 per cent for periods up to one hour is described. The sampled signal is connected to one input of a diode-type electronic switch with a feedback loop from the output to the second input. The switching signal consists of a train of equally-spaced pulses. When the switching pulse is applied, the feedback loop is opened and the input is connected to the output, charging a capacitor. The switch is a current-limiting device, providing a constant charging current. Removing the switching pulse closes the feedback loop, and the output holds the level of the input signal at the time of switching. The spacing between the switching pulses is short compared to the time constant of the closed-loop circuit. The limiting factor in achieving extremely long holding times (calculated values are 10^4 hr) is imperfect balancing of the amplifier.

Selective Erasure and Nonstorage Writing in Direct-View Halftone Storage Tubes—*see* 1501.

1617
An Introduction to Analogue Computer Methods by J. G. Thomason (Imperial Chemical Industries, Ltd.); *Computer J.*, vol. 3, pp. 211–219; January, 1961.

A basic tutorial paper on the main procedures and applications of analog computation is presented. The simplicity of analog programming makes the analog computer an ideal tool for simulation studies. Few advances in analog techniques are expected, but many fields of application remain to be explored. The main disadvantage is the lack of accuracy of the analog compared with the digital computer.

1618
Initial Conditions in Computer Simulation by K. S. Miller (New York University) and J. B. Walsh (Columbia University); IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-10, pp. 78–80; March, 1961.

A technique for the straightforward simulation of the transfer function of a certain class of linear systems is developed. This method is particularly well adapted to the analysis of systems with fixed transfer functions and variable initial conditions and forcing functions. In particular, a single simulation, minimal in its use of integrators, will suffice to handle forcing functions and initial conditions on both input and output.

Conformal Transformations by Analog Computer—*see* 1566.

1619
Design and Operation of the Ceilometer Computer by P. Meissner (NBS); *U. S. Govt. Res. Repts.*, vol. 35, p. 82 (A); January 13, 1961. PB 161 565 (order from OTS $2.00)

The ceilometer computer, which has been developed to provide a display and electrical readout of cloud-height information for use with an automatic weather station, is described. The computer receives an analog signal from the detector of a rotating-beam ceilometer, and determines the height at which cloud indications occur. A small magnetic storage drum contains 10 minutes of cloud-height information which is continuously updated, and these data are analyzed for the following factors: 1) predominant cloud height over the past 10 minutes, 2) maximum and minimum height at which significant cloud occurrences were observed, and 3) number of cloud observations up to a selected critical altitude. The computer is a wired-program machine constructed of transistorized plug-in packages. Several modes of manual operation have been incorporated for testing and maintenance purposes.

1620
A High-Speed Analogue to Digital Convertor by N. Winterbottom and J. S. B. Walters (Armstrong Whitworth Aircraft Ltd.); *Electronic Engrg.*, vol. 33, pp. 144–149; March, 1961.

A high-speed analog to digital convertor capable of performing 55,000 conversions per second from an analog voltage to an 8-bit binary code is described. The equipment uses transistors as the active elements.

12. REAL-TIME SYSTEMS AND AUTOMATIC CONTROL; INDUSTRIAL APPLICATIONS

1621
A Simulator Study of a Two-Parameter Adaptive System—by R. J. McGrath and V. C. Rideout (University of Wisconsin); IRE TRANS. ON AUTOMATIC CONTROL, vol. AC-6, pp. 35–42; February, 1961.

The use of sinusoidal parameter perturbation applied to a feedback control system to obtain an adaptive scheme which optimizes the system for changes in inputs and/or system parameters is discussed. It is shown that if a parameter perturbation signal is cross correlated with the system error

squared, the correlator output can be used to adjust the parameter to minimize the mean-square error. Other error measures may also be used. Two or more parameters may be simultaneously adjusted if they are perturbed at different frequencies, and each provided with an independent adaptive loop. A computer simulation of a third-order system having two adjustable parameters has been examined for a variety of inputs including random signals. It is shown that the scheme minimizes the mean-square error in all cases.

1622
Time-Optimal Control of Higher-Order Systems by F. B. Smith, Jr. (Minneapolis-Honeywell Regulator Co.); IRE TRANS. ON AUTOMATIC CONTROL, vol. AC-6, pp. 16–21; February, 1961.

Practical extension of time-optimal control to systems of higher order than three has been limited primarily by difficulties in physically representing surfaces in a phase space of these higher dimensions. A method for obtaining the forcing function as a function of the state variables without requiring use of the phase space concept is presented. On-line solution of a set of transcendental equations is required. Results of a digital simulation of a fourth-order, real-root, single-degree-of-freedom system are presented. In a digital solution the system operates as a series of short open-loop control intervals. The effect of including derivatives of the input for prediction is shown for second-order model inputs.

1623
An Optimal Strategy for a Saturating Sampled-Data System by C. A. Desoer and J. Wing (University of California); IRE TRANS. ON AUTOMATIC CONTROL, vol. AC-6, pp. 5–15; February, 1961.

The usual sampled-data control system in which the sampler is followed by a zero-order hold and the transfer function is $G(s)=1/s(s+a)$ is considered. Saturation is represented by the fact that the forcing function applied to $G(s)$ may not be larger than 1 in absolute value. The problem is to determine the saturating zero-order hold forcing function which forces the system from an arbitrary initial state to equilibrium in the least number of sampling periods. Such a forcing function is defined as an optimal strategy. The state plane is divided into boundary states and interior states. To each boundary state corresponds a unique optimal strategy. To each interior state correspond infinitely many optimal strategies. From the system parameters a polygonal curve, called the critical curve, is defined in the state plane. An optimal strategy is then proposed in which the required forcing function is simply obtained by computing distance of the representative point in the state plane to the critical curve. A simple computer is proposed to implement this optimal strategy. Finally, the proposed optimal strategy is shown to reduce in the limit as $T\rightarrow 0$ to that of the corresponding continuous system.

1624
Discrete Analogue-Computer Compensation of Sampled-Data Control Systems by T. Glucharoff (University of New South Wales); *Proc. IEE*, vol. 108B, pp. 167–179; March, 1961.

An analog computer which can solve pulse transfer functions and operate as a discrete controller for compensation of sampled data or continuous systems is described. Operational amplifiers and silicon-diode switches are combined to perform the basic functions of sampling, holding and time delay. The accuracy obtainable is comparable with that of digital controllers, but the analog computer has the advantages of simplicity of setting-up and low cost. Adjustment of the controller parameters is easy, and the computer can be used to determine the coefficients of the pulse transfer function for optimum operation before a digital computer is employed. A simplified method of discrete-controller design for saturating sampled-data systems is also presented.

1625
GENDARE System: Fundamental Concepts of Logic and Structure by J. B. Williams, Jr. (Lincoln Lab., M.I.T.); *U. S. Govt. Res. Repts.*, vol. 35, p. 323 (A); March 10, 1961. PB 152 722 (order from LC mi $2.40, ph $3.30).

GENDARE (*Gen*eralized *Da*ta *Re*duction), a programming system to provide data reduction services easily and cheaply when large volumes of data are to be processed, is described. GENDARE design starts with the observation of logical redundancies in the class of programs devoted to data reduction. On this basis, the existence of a general case for this program class is postulated, assumptions are made about the form of the inputs, and determining characteristics for the processing to be performed are discussed. From this foundation, the fundamental elements of GENDARE logic and structure are developed as a natural consequence.

1626
Automatic Processing System for Acoustical Data by W. E. Parker and L. V. East (Boeing Airplane Co.); *J. Acoust. Soc. Am.*, vol. 33, pp. 1–6; January, 1961.

An improved instrumentation system developed to analyze the dynamic pressure field produced by jet aircraft is described. The system is automatic and utilizes analog computer techniques to obtain an accuracy of ± 0.2 db in its readout. The method used is to secure a true rms value of octave band segments of the jet noise spectrum and to convert this to decibels by the use of a dc logarithmic amplifier. The transition to acoustical decibels is made by obtaining the difference between this voltage and another voltage obtained from a reference oscillator. The value of this reference oscillator in acoustical decibels is obtained by comparing its output voltage to that of a calibrated microphone. The dynamic range of the system has proved more than adequate for greater than 98 per cent of the data processed by this unit during the time this system has been in operation. The output is in the form of punched cards for utilization with a digital computer.

Advanced Computer Applications—*see* 1533.

1627
Computer Production Control—The Second Year by D. L. J. Hughes (Int. Computers and Tabulators, Ltd.); *Computer J.*, vol. 3, pp. 198–201; January, 1961.

A computer program to schedule the production of a small factory is described. Experience gained on an earlier program was used to improve performance. The program is divided into successive stages of scheduling and breakdown. There are seven such stages between the most complex assemblies and raw materials, necessitating four computer runs. The resulting high accuracy justifies the computer time required.

13. GOVERNMENT, MILITARY, AND TRANSPORTATION APPLICATIONS

1628
Generalized Simulation of Post Office Systems by R. C. Brigham and P. D. Burgess (Radiation Inc.); *J. Assoc. Comp. Mach.*, vol. 8, pp. 252–259; April, 1961.

An IBM 704 program which simulates mail processing systems within any United States Post Office is described. Such systems can be represented by a connection of basic processing operations, of which there are only four different types. Any desired connection is possible, and therefore any size postal system can be simulated. The basic processing calculations are given and the program flow chart is briefly discussed. Some of the applications and results are indicated.

1629
Torpedo Hit Probabilities by E. S. Wolk (University of Connecticut and Genl. Dynamics); *SIAM Rev.*, vol. 2, pp. 292–296; October, 1960.

An analytic method for computing a torpedo hit probability in terms of the quantities available to the dispatcher of the torpedo is described. A feature of the method is that the estimates of all the parameters relating to target motion are assumed to be in error, and that these errors are not considered to be independent. The technique has been programmed and run on an IBM 704.

1630
Techniques for Producing School Time Tables on a Computer and their Application to other Scheduling Problems by J. S. Appleby D. V. Blake, and E. A. Newman (Natl. Physical Lab.); *Computer J.*, vol. 3, pp. 237–245; January, 1961.

The logical problems encountered in preparing a high-school classroom schedule and a typical manual means of solving them are described. Techniques for adapting the problem to computer solution are discussed. Essentially, the computer searches for a solution by filling slots one at a time until it either reaches a conclusion or a contra-

diction. An improved version of the program is predicted to be 200–400 times as efficient as a human. Related problems to which the techniques apply are factory scheduling and air traffic control.

An Electronic Analog Computer for a Coal Transportation Problem—*see* 1612.

1631
An Investigation of Real-Time Solution of the Transportation Problem by R. Totschek and R. C. Wood (Systems Development, Inc.); *J. Assoc. Comp. Mach.*, vol. 8, pp. 230–239; April, 1961.

An investigation of methods for solving the transportation problem within the framework of a cyclic real-time digital computer program system is reported. Two techniques for obtaining a basic feasible solution, the Northwest Corner Rule and a modification of Vogel's Approximation Method, are considered. The solution is then iterated using the Dual Theorem. A statistical analysis of the solution times of each method is made for different matrix dimensions. Methods for truncating the process are considered and recommendations are made. The use of Vogel's Approximation Method is shown to yield significantly faster solution times.

1632
Programming IBM 650 RAMAC Computer for Data Processing in an Air Route Traffic Control Center by H. R. Johnson, Jr., J. E. Erickson, *et al.* (Natl. Aviation Facilities Experimental Ctr.); *U. S. Govt. Res. Repts.*, vol. 35, p. 39 (A); January 13, 1961. PB 171 040 (order from OTS $2.50).

The development and preparation of a detailed ATC program for an IBM-650/RAMAC computer system are described. Many portions of this program are predicated on the ability of this system to process all of the many types of flight plan information, to produce flight progress strips, and furnish other information needed for the control of air traffic at the Indianapolis Air Route Traffic Control Center. A description of the types and quantities of flight plans which must be processed is given. A general discussion of the program logic used for processing airway and/or direct route flight plans is included and is illustrated by charts, flow diagrams, and pictures.

14. BUSINESS APPLICATIONS

1633
The Experience of Applying a Commercial Computer in a British Organization by A. J. Platt (Pilkington Bros. Ltd.); *Computer J.*, vol. 3, pp. 185–197; January, 1961.

The detailed procedure, from the initial decision to purchase a computer to the final checking out of programs for sales, materials and wage problems, is described. The purchase of the computer was initially justified by its projected performance on sales alone. Its subsequent application to materials-mix, research and payroll were regarded as bonuses. The ratio of time taken to delineate a computer task to time taken to program the computer for the task was found to be of the order of four to one.

1634
PERT: Program Evaluation and Review Technique for the Royal McBee LGP-30 Computer (Army Chem. Ctr.); *U. S. Govt. Res. Repts.*, vol. 35, p. 80 (A); January 13, 1961, PB 150 992 (order from LC mi$4.80, ph$13.80).

The results of a study undertaken to convert the Booz-Allen-Hamilton Program Evaluation and Review Technique (PERT) for small-scale applications to the Royal McBee LGP-30 computer are reported. The program function and capabilities, operating instructions, LGP-30 flow chart, Symbolic and machine language codes, data input error check list, data tables and LGP-30 subroutine and data table assembly are all discussed.

SUBJECT INDEX

AUTHOR INDEX

PGEC News

To All PGEC Chapter Officers and Committee Chairmen

The PGEC *News* section of the TRANSACTIONS is open for your announcements and reports of activities. Deadline is the first of the month, two months ahead of the date of issue. Send all items to the *Editor*.

THE CHAIRMAN'S LETTER

St. Paul, Minn., July 15, 1961—

Editorial Shift

This is my first opportunity (because of fantastically long lead time imposed on the chairman) to welcome our new Editor-in-Chief, Professor Norman R. Scott of the University of Michigan, whose appointment was acted upon at the May meeting of the PGEC AdCom. His biography and picture appeared in the June issue. Norm is one of the top-drawer workers in the electronic computer field and is singularly well qualified for the editorship. We wish him much success in carrying out this most important mission to our membership and to the electronic computer community.

To Dr. Howard Tompkins, our retiring Editor, I wish to express the commendation and the gratitude of the PGEC AdCom for the tremendous job he has done during his two and one-half years in the Editor's chair. He has demonstrated outstanding leadership and organizational skill in the development and expansion of the TRANSACTIONS. During his tenure, the reviews section, associate editorships, and special issues were instituted. He has developed an impressive list of competent referees to help him maintain the highest standards of editorial excellence. We wish Howard the best of success in his new career, and hope that PGEC will continue to enjoy the benefit of his abilities and his wise counsel.

AFIPS

The American Federation of Information Processing Societies (AFIPS) began its existence on May 10, 1961, at the final meeting of the National Joint Computer Committee (NJCC). The AFIPS governing board consists of four directors from each of the three member societies. At the first meeting of the board, Dr. Willis H. Ware was elected chairman. Willis is well known as a past chairman of PGEC, and for his long participation and leadership in the activities of the computing community. AFIPS directors for IRE are: Werner Buchholz, Frank E. Heart (who was subsequently appointed AFIPS treasurer), Harry T. Larson, and myself. The chairmen of all three groups representing the member societies are on the initial board. These men, together with the AFIPS chairman, constitute the AFIPS executive committee.

Committees have been established in AFIPS for setting up bylaws and policy on finances, admission of new member societies, conferences, publications, public relations, etc. These groups are all busy at this writing.

The constitution of the new Federation appears elsewhere in this section.

Administrative Committee

The AdCom is currently deliberating a number of policy matters, and I expect that there will be significant results to report in my next letter. Dick Melville heads a special committee on Structure Study, which is looking into the desirability of establishing special interest subgroups or divisions within PGEC. The main objective is to facilitate conference and symposium activity in specialized technical areas. The Constitution and Bylaws Committee, under Roger Sisson, is studying necessary changes and additions in the light of our current objectives. One of this committee's priority jobs is to generate suitable bylaws governing our representation and financial relationships with AFIPS. Such bylaws must necessarily interlock with the AFIPS bylaws now being developed.

New objectives, and the means for their vigorous implementation, are currently being worked out for the Membership Committee, Chapter Activities Committee, and Student Activities Committee. This work is currently in process and will be reported on next time.

I am happy to report that our Secretary-Treasurer, Pete Zimmer, is now fully recovered and has returned to this post after several months' leave. I want to express the thanks of the AdCom to Chuck Pallas, who was acting Secretary-Treasurer during Pete's absence, for his wonderful cooperation and for the excellent job which he did during this period.

The next meeting of the AdCom, scheduled for August 23 at San Francisco in connection with WESCON, will have taken place by the time this issue appears. The meeting after that will be in Washington in December, at the EJCC.

ARNOLD A. COHEN
Chairman, PGEC

CONSTITUTION OF THE AFIPS

ARTICLE I. NAME

The name of this organization shall be The American Federation of Information Processing Societies (Unincorporated), hereinafter known as the Federation.

ARTICLE II. PURPOSES

The purposes of this Federation shall be the advancement and diffusion of knowledge of the information processing sciences and for literary and scientific purposes within the meaning of Section 501(c)(3) of the Internal Revenue Code of 1954. These sciences include, but are by no means restricted to, the computer sciences and their applications to society. To this end, it is part of the purpose of this Federation, among other measures, to serve the public by making available to journals, newspapers, and other channels of public information reliable communications as to information processing and its progress; to cooperate with local, national, and international organizations or agencies on matters pertaining to information processing; to serve as representative of the United States of America in international organizations with like interests; to promote unity and effectiveness of effort among all those who are devoting themselves to information processing by research, by application of its principles, by teaching or by study; and to foster the relations of the sciences of information processing to other sciences and to the arts and industries. In pursuing these purposes, the Federation shall do nothing that is in direct competition with activities of its member societies and no substantial part of its activities shall be carrying on propaganda, or otherwise attempting to influence legislation, or participating in, or intervening in (including the publishing or distributing of statements), any political campaign on behalf of any candidate for public office.

ARTICLE III. CLASSES OF MEMBERSHIP

1. Member Society: The Member Societies of the Federation shall be the following founding Societies:

American Institute of Electrical Engineers
Association for Computing Machinery
Institute of Radio Engineers

and such other nonprofit corporations and unincorporated associations composed of individual voting members as may be elected Member Societies as provided in Article V and which meet the following requirements:

a) Operate principally in the United States of America.
b) Have a nationwide field of operation.
c) Be devoted to the advancement of the purposes of the Federation.
d) Be composed of, or have a subdivision composed of, at least 400 members who are professionally engaged in the information processing sciences or applications thereof.
e) Be independent of all commercial, industrial or other profitmaking or labor organizations.

2) Associate Member Society: An Associate Member Society shall be any organization composed of individual voting members as may be elected as Associate Member Society as provided in Article V and which meets the following requirements:

a) Operates principally in the United States of America.

b) Has a nationwide field of operation.
c) Is composed of, or has a subdivision composed of, at least 400 members devoted to the advancement of the purposes of the Federation, who have interests closely related to the information processing sciences or applications thereof.
d) Is independent of all commercial, industrial, or other profit-making or labor organizations.

3) Affiliated Society: An Affiliated Society shall be any organization which is interested in information processing, and which is elected as provided in the Bylaws.

4) Sustaining Associate: A Sustaining Associate shall be any corporation, institution or individual interested in information processing who annually pays dues as fixed by the Bylaws, and who is elected as provided in the Bylaws.

Article IV. Meetings of the Federation

1) An annual meeting of the Federation shall be held at a time and place set each year by the Chairman of the Governing Board for the purposes of electing Member Societies or Associate Member Societies or dropping them from membership as provided herein, and of acting on Constitutional amendments and for such other purposes as may be stated in the notice of meeting.

Special meetings of the Federation may be called by the Chairman or by the Secretary upon resolution of the Governing Board or upon the written request of at least two-fifths of the Member Societies for the purposes stated in the preceding paragraph.

Notices of annual and special meetings of the Federation shall be sent by the Secretary or other officer to each Member Society not less than fifteen (15) nor more than forty (40) days before the meeting, directed to each Member Society at its address as it appears in the records of the Federation. The notice of a special meeting shall state the purpose for which the meeting is called.

2) Only Member Societies shall be entitled to vote at any such meeting of the Federation, and each Member Society shall be entitled to one vote at such meetings.

3) Representation of a Member Society shall be certified in writing by the Secretary of the Member Society.

4) At any meeting of the Federation, the presence of proxies representing a majority of the Member Societies shall constitute a quorum.

Article V. Election to Various Classes of Membership

Nominations for the election of a Member Society or Associate Member Society shall be made by the Governing Board. Notice of such nominations shall be given by the Secretary of the Federation to each Member Society, and these nominations shall be voted upon by the Member Societies at a meeting of the Federation held not less than six months or more than one year after such notice. The election of a Member Society or Associate Member Society shall require the affirmative vote of not less than three-fourths of the Member Societies.

Article VI. Dropping From Membership

1) When the Governing Board shall determine that a Member Society or Associate Member Society has ceased either to devote itself to the advancement of the Federation's purposes or, for not less than two years, to perform the duties of membership, the Board may recommend that such society be dropped from membership. A Member Society or Associate Member Society shall be dropped from membership upon the affirmative vote of at least three-fourths of the Member Societies. The vote is to be taken at a meeting of the Federation held not less than six months nor more than one year after notice of such a recommendation has been given to the Member Societies.

2) An Affiliated Society or Sustaining Associate may be dropped at any time in such manner as the Bylaws shall provide.

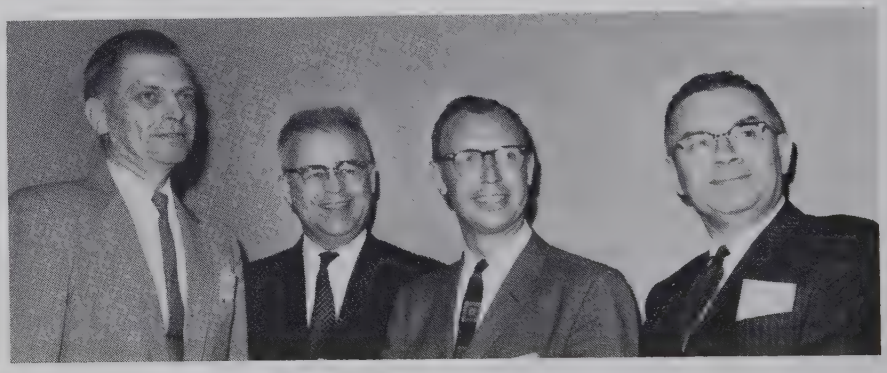
3) Any Member Society, Associate Member Society, Affiliated Society or Sustaining Associate may resign by notifying the Governing Board in writing signed by the Secretary of the resigning society.

Federation that he is no longer a member of the society.

In case any director shall die, resign, become incapacitated or disqualified to act as such, or his position becomes otherwise vacant, the Member or Associate Member Society which designated him shall be entitled to designate a successor to hold office for the unexpired portion of the term of office of the director whose place shall have become vacant.

3) A director may appoint in writing a proxy to represent him at a meeting of the Governing Board, provided that the appointed individual is a member of the society that the director represents.

4) Should a director or an officer resign at any time, he shall present his resignation in writing to the Secretary of the Federation and it shall take effect immediately unless a future date is named in the resignation. The Secretary's resignation shall be presented to the Chairman. No acceptance of a resignation shall be necessary, but immediate notice of a resignation shall be



Executive committee of the newly-formed American Federation of Information Processing Societies are (left to right): R. A. Imm, IBM Corporation, representative of the American Institute of Electrical Engineers; Dr. Harry Huskey, University of California, representative of the Association for Computing Machinery; Dr. Willis H. Ware, Computer Sciences Department, RAND Corporation, Chairman of AFIPS Governing Board; and Dr. A. A. Cohen, Remington Rand Univac, representative of the Institute of Radio Engineers.

Article VII. Governing Board

1) The business and activities of the Federation, except as specified in Article IV, shall be managed by its board of directors which shall be called collectively the Governing Board. The Governing Board shall be comprised of the total number of directors as determined in Section 2 of this Article. Each director shall have one vote on the Governing Board, except as provided in Section 5 of this Article.

2) Each Member Society shall be entitled to designate four directors as follows:
Two in even-numbered years.
Two in odd-numbered years.
Each Associate Member Society shall be entitled to designate one director.

A director shall be designated for a term of two years and shall serve until his successor is designated and seated. The term of each director shall begin at the close of the annual meeting of the Federation following his designation.

A director designated by a Member or Associate Member Society shall cease to be a director of this Federation whenever said Society shall certify to the Secretary of the given to the Member or Associate Member Society by which he was designated.

5) A director who is not a citizen of the United States, unless he has filed a declaration of intention to become a citizen, cannot enter motions or vote on matters pertaining to representation of the United States of America in international organizations.

Article VIII. Meetings of the Governing Board

1) Meetings of the Governing Board shall be called in such a manner as is provided in the Bylaws.

2) A majority of the members of the Governing Board shall constitute a quorum for the transaction of business, including the establishments and amendment of Bylaws.

Article IX. Finances

1) Dues: Each member organization will contribute annual dues as follows: Member Societies will pay equal amounts. The amount is to be ratified by an affirmative vote of not less than three-fourths of the Member Societies at a meeting of the Federation and remains in effect until

changed. Associate Member Societies, Affiliated Societies and Sustaining Associates will pay annual dues as prescribed in the Bylaws.

2) Dissolution: In the event of dissolution of the Federation, the surplus assets, if any, shall be distributed to those member and associate member societies which are themselves organized for tax-exempt purposes and in proportion to the total amount of dues paid in by them to the Federation.

3) Budget: An annual budget to carry out the operation of the Federation shall be prepared and submitted to the Member Societies at least ninety days prior to the start of each fiscal year of the Federation.

Article X. Services

When consistent with its educational and scientific purposes, the Federation shall provide such services to individual member organizations as may be authorized by the Governing Board, and these services shall be self-supporting. Terms and conditions under which services are to be performed shall be established by agreement between the Federation and the organization served.

Article XI. Officers

1) The officers of the Federation shall be Chairman of the Governing Board, Secretary, Treasurer, a principal executive officer and such other officers as may be provided for in the Bylaws or by resolution of the Governing Board. Officers serving without compensation shall be elected annually by the Governing Board. Compensated officers shall serve at the pleasure of the Board.

2) The Chairman shall be elected from among the outgoing or former directors.

3) The duties of the officers shall be those usually devolving upon such officers except as may be otherwise provided in the Bylaws or by resolution of the Governing Board.

4) The Chairman shall not hold a position of director while holding the office of Chairman.

5) If the Chairman is unable to preside at a meeting of the Governing Board, an Acting Chairman shall be elected *pro tem* by the Governing Board.

Article XII. Committees

The Governing Board shall appoint an Executive Committee from among all directors, subject to the condition that a majority shall be from the directors designated by Member Societies as defined in Article VII. The Executive Committee shall consist of the Chairman of the Governing Board and at least one representative of each Member Society. The Executive Committee shall be charged with the conduct of such business as may be assigned to it by the Governing Board.

The Governing Board or the Executive Committee shall have power to create and assign duties to other committees. The members of such committees may or may not be members of the Governing Board. The Chairman of the Governing Board shall be an *ex-officio* member of all committees.

Article XIII. Annual Report

The Governing Board shall report in writing annually to the member organizations as to the activities of the Federation in the year for which the report is made, and shall include in the report a statement of its financial condition.

The Governing Board shall also make such other reports as are required by law.

Article XIV. Amendments to Constitution

1) An amendment to the Constitution must be proposed to the Governing Board by a member of the Board. If then it receives the approval of the Board by a majority vote of the full membership of the Board, the proposed amendment shall be submitted to the Member Societies and shall be adopted upon the favorable vote of not less than three-fourths of the Member Societies which take action on the submitted amendment within a period of not more than one year after notice of the proposed amendment is officially sent to them, provided that in every case the amendment has had the approval of not less than three-fourths of the Member Societies.

2) The vote of a Member Society on an amendment duly proposed shall be evidenced to the Federation by a written certificate signed by such officers of the Member Society as the governing body of that society may specify.

Article XV. Interpretation of the Constitution

The Governing Board shall be empowered to interpret the Constitution and Bylaws by majority vote of the entire Governing Board.

NEW ASSOCIATE EDITOR

Edward J. McCluskey, Jr. (S'51–M'55), whose appointment as Associate Editor for Logic and Switching Theory is announced on page 345, is well known to readers of this journal as its first Reviews Editor and also as a frequent contributor. He was born in Brooklyn, N. Y., on October 16, 1929. He received the A.B. degree in mathematics and physics from Bowdoin College, Brunswick, Me., in 1953, and the B.S., M.S., and Sc.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1953 and 1956, respectively.

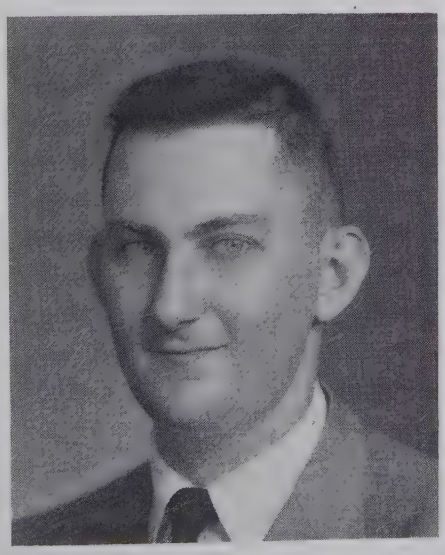
From 1953 to 1955, he was a research assistant and instructor at M.I.T. In 1955, he joined the staff of Bell Telephone Laboratories, Whippany, N. J., where he did research in connection with electronic switching systems. He also served as a Lecturer at the College of the City of New York. He became Associate Professor of electrical engineering at Princeton University, Princeton, N. J., in September, 1959.

Dr. McCluskey is a member of Phi Beta Kappa, Tau Beta Pi, Eta Kappa Nu, Sigma Xi, and the Association for Computing Machinery. His many outstanding papers on logic and switching theory demonstrate his excellent qualifications as Associate Editor for this important area.

NEW REVIEWS EDITOR

Thomas C. Bartee (M'57), whose appointment as Reviews Editor is announced on page 345, has served as an Assistant Reviews Editor and has also contributed papers of his own. He was born on December 18, 1926, in Moberly, Mo. He received the A.B. degree from Westminster College, Fulton, Mo., in 1949. From 1944 to 1946, he served in the U. S. Navy and attended Naval Electronics schools at Chicago, Ill., Del Monte, Calif., and San Francisco, Calif., following which he served in the South Pacific.

From 1953 to 1956, he was a member of the engineering staff of the Firestone Guided Missile Division, Los Angeles, Calif., engaged in design and development of electronic guidance systems for missiles. Since 1956, he has been a staff member of the Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, where he is



E. J. McCluskey, Jr.

T. C. Bartee

currently directing studies of logical design techniques. He is the author of a book "Digital Computer Fundamentals" published by McGraw-Hill Book Co., Inc., New York, N. Y., and co-author, with Irving Reed and Irwin Lebow, of the book "Theory and Design of Digital Machines," which will appear soon. He is co-editor, with E. J. McCluskey, Jr. of the book "A Survey of Modern Switching Theory," which is now in preparation.

Mr. Bartee is a member of PGEC and of PGIT. He is also a member of the AIEE Computing Devices Committee and of the Logic and Switching Circuits Subcommittee.

CHAPTER ACTIVITIES

The Detroit Chapter has completed a busy year of activities under its 1960–1961 Officers:

Chairman—J. L. McKelvie, Bendix Research Laboratories, Detroit
Vice Chairman—George Zacharopoulos, Lafayette Clinic, Detroit
Secretary—Yuji Morita, Institute of Science and Technology, The University of Michigan, Ann Arbor

Four meetings were held as follows:

1) November 1, 1960, Detroit: "The Application of Tunnel Diodes to Digital Computers," by F. K. Buelow, IBM Product Development Laboratories, Poughkeepsie, N. Y.
2) November 28, 1960, at the University of Michigan's Willow Run Research Laboratories: "Modeling Techniques for Digital Computers," by Richard R. Legault, Institute of Science and Technology, The University of Michigan.
3) January 30, 1961, at Lafayette Clinic, Detroit: "The Use of Automatic Data Reduction Techniques in the Study of Physiological Aspects of Emotion," by Dr. A. F. Ax and G. Zacharopoulos, of the Lafayette Clinic.
4) May 1, 1961, at the Cooley Building, The University of Michigan, Ann Arbor: "Adaptive Mechanisms and Simple Learning Machines," by Prof. F. H. Westervelt, The University of Michigan.

During the past year, both the Association for Computing Machinery and the American Institute of Electrical Engineers substantially increased their computer-oriented activity in the Detroit area. Local members of the ACM formed a Metropolitan Detroit Chapter and held meetings during the winter and spring. The AIEE Michigan Section established an Electronic Computer Committee late in the spring. Both of these moves were encouraged by the PGEC Executive Committee as constructive steps towards enhancing Detroit's position and image as a major center of computer activity.

A direct consequence of these developments has been the decision to hold the 1963 Spring Joint Computer Conference in Detroit. The choice was made by the Executive Committee of the newly formed American Federation of Information Processing Societies, successor to the National Joint Computer Committee. The decision resulted from representations made to the committee by the Detroit ACM, AIEE, and IRE organizations, all of which had to have formal local chapters to qualify Detroit as a potential site for a Joint Computer Conference.

Plans for 1961–1962

The following officers have been chosen for 1961–62:

Chairman—G. Zacharopoulos
Vice Chairman—Y. Morita
Secretary—W. E. Chapelle, Bendix Research Laboratories, Detroit

Plans for next year are currently being prepared. A significant new feature will be the close cooperation of the PGEC chapter and the local AIEE Electronic Computer Committee. Some joint meetings will be held, and the two societies will cooperate in publicizing separate meetings. It is hoped that similar arrangements can be made with the ACM Chapter so that a program can be developed to suit the wide range of interests among computer people in the Detroit area. The PGEC Chapter also hopes to set up a number of informal Technical Panels for exchange of ideas among specialists in several major areas of interest.

PGEC PEOPLE

Roger L. Sisson (S'48–A'50–M'51), noted management consultant in the field of information technology, will direct advanced programs at Auerbach Electronics Corporation, it was announced by Isaac L. Auerbach, President of the Company.

He will be responsible for initating new programs and directing special projects in the field of information technology. He also will serve as a technical consultant and advisor on management systems projects and long-range corporate planning.

Previously, Mr. Sisson was Manager of Program Analysis at Aeronutronic, a division of Ford Motor Company, where he also directed system design and programming for the Army Tactical Operations Central. From 1954 to 1958, he was a partner in the management consulting firm of Canning, Sisson and Associates, working on all facets of electronic data processing and business systems. He analyzed computer market potentials and designed production control, accounting, and inventory control systems for manufacturing, railroad, and retail firms.

Prior to that time, he was with the National Cash Register Corporation as Manager of Customer Computing Services, an activity which he organized. He also served as an engineer with this firm and earlier with Electronic Engineering Company, designing, developing and testing data processing equipment.

R. L. Sisson

An active lecturer and writer, Mr. Sisson founded the *Data Processing Digest*, is co-author of a book on management decision simulation, and has published extensively in technical publications. He has served as a Lecturer in the Business School of the University of Southern California, Los Angeles, and has performed operations research studies for the University of California, at Los Angeles.

He received the B.S. and M.S. degrees in electrical engineering from Massachusetts Institute of Technology, Cambridge. Mr. Sisson is currently a member of the Administrative Committee of the IRE-PGEC, and is currently serving as Chairman of the Constitution and Bylaws Committee of the PGEC. He has been active in the Association for Computing Machinery, The Institute of Management Sciences, and the Operations Research Society of America.

Notices

This *Notices* Section is open to all who have an announcement of a conference, symposium, session, publication, or other artifact of interest to the PGEC membership. Please send announcements to the *Editor*, who will put them in the first available issue. The right is reserved to edit the announcements, and to decide whether they indeed are aimed at our audience.

COMPUTER APPLICATIONS SYMPOSIUM—ARMOUR RESEARCH FOUNDATION

The 1961 Computer Applications Symposium sponsored by Armour Research Foundation will be held October 25 and 26 at the Morrison Hotel, Chicago.

Invited papers will represent business and management, and engineering and scientific applications. The conference will stress user experience in computer application and programming techniques.

Chairman for the business and management applications session, October 25, is Charles A. Phillips, director, Data Systems Review Division, Office of the Assistant Secretary of Defense, Washington, D. C. Dr. Robert P. Rich, director, Computation Facility, Applied Physics Laboratory, The Johns Hopkins University, Silver Springs, Md., is chairman for the scientific and engineering applications session, Ocotber 26.

Inquiries concerning the conference should be addressed to Benjamin Mittman, Conference Program Chairman, Armour Research Foundation, 10 W. 35th St., Chicago 16, Ill.

CONFERENCE ON NONLINEAR MAGNETICS

The 1961 Special Technical Conference on Nonlinear Magnetics is scheduled for November 6–8 at the Statler Hilton Hotel in Los Angeles. This meeting is being sponsored by the IRE Professional Group on Electronic Computers, the Professional Group on Industrial Electronics, and the AIEE. For further details, contact Dr. Ted Bernstein at Space Technology Laboratories, P. O. Box 95001, Los Angeles 45, Calif.

COMING MEETINGS—PAPERS DEADLINE PAST

AIEE FALL MEETING

The AIEE will again sponsor a Symposium on Switching Circuit Theory and Logical Design at its Fall General Meeting, to be held in Detroit, Mich., during the week October 15–20, 1961, at the Statler-Hilton Hotel.

The program is as follows:

Tuesday, October 17

Symposium Chairman: R. S. Ledley
National Biomedical Research Foundation, Silver Spring, Md.

Welcome Address: R. S. Ledley.

Opening Remarks: T. H. Mott, Jr., Lockheed Electronics Company, Bedminster, N. J.

Invited Address: Howard Aiken, Harvard University, Cambridge, Mass

MINIMIZATION OF BOOLEAN FUNCTIONS

Chairman, T. H. Mott, Jr.

"On Application of Linear Programming to the Minimization of Boolean Functions," *A. Cobham, R. Fridshal, and J. H. North*, IBM Research Center, Yorktown Heights, N. Y.

"Minimal Sums for Boolean Functions Having Many Unspecified Fundamental Products," *E. J. McCluskey, Jr.*, Department of Electrical Engineering, Princeton, University, Princeton, N. J.

"Use of a List Processing Language in Programming Simplification Procedures," *S. R. Petrick*, Electronics Research Directorate, Air Force Cambridge Research Laboratories, Bedford, Mass.

Wednesday, October 18

THRESHOLD LOGIC DESIGN

Chairman, C. C. Elgot
IBM Research Center
Yorktown Heights, N. Y.

"Threshold Logic and Two-Person Zero-Sum Games," *S. B. Akers, Jr.*, Electronics Laboratory, General Electric Company, Syracuse, N. Y.

"On the Characterization of Threshold Functions," *C. K. Chow*, Burroughs Corporation, Paoli, Pa.

"Functional Forms of Majority Functions and a Necessary and Sufficient Condition for their Realizability," *S. Muroga*, IBM Research Center, Yorktown Heights, N. Y.

"The Profile Technique for the Design of Threshold Device Logic," *O. B. Stram*, Burroughs Corporation, Burroughs Laboratory, Paoli, Pa.

"More About Threshold Logic," *R. O. Winder*, RCA Laboratories, Princeton, N. J.

NEURON PHYSIOLOGY AND THRESHOLD LOGICS

Chairman, R. S. Ledley

"The Integrative Properties of Neurons," *D. Kennedy*, Department of Biological Sciences, Stanford University, Stanford, Calif.

"Mathematical Models of Neuron Interaction" *H. D. Landahl*, Committee on Mathematical Biology, The University of Chicago, Chicago, Ill.

"Multivalued Logic Devices for Simulating Threshold Neurons," *D. R. Boyle and R. S. Ledley*, National Biomedical Research Foundation, Silver Spring, Md.

"Many Valued Logics and Reliable Homeostatic Processes," *J. D. Cowan*, Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, Mass.

Thursday, October 19

INTRODUCTION TO SPEED INDEPENDENT CIRCUITS

Session arranged by D. E. Muller, Digital Computer Laboratory, University of Illinois, Urbana, Ill.

Chairman, S. Muroga

"An Introduction to Speed Independent Circuit Theory," *R. E. Miller*, IBM Research Center, Yorktown Heights, N. Y.

"One Method for Designing Speed Independent Logic for a Control," *R. E. Swartwout*, Digital Computer Laboratory, University of Illinois, Urbana, Ill.

"Problems in the Physical Realization of Speed Independent Circuits, *J. E. Robertson*, Digital Computer Laboratory, University of Illinois, Urbana, Ill.

"A Flow Chart Notation for the Description of a Speed Independent Control," *D. B. Gillies*, Digital Computer Laboratory, University of Illinois, Urbana, Ill.

SAMUEL H. CALDWELL MEMORIAL SESSION
FINITE AUTOMATA

Chairman, E. J. McCluskey, Jr.

"S. H. Caldwell, 1890–1960, *E. J. McCluskey, Jr.*

"Transient Behavior in Iterative Combinational Switching Networks," *W. L. Kilmer*, Electrical Engineering Department, Montana State College, Bozeman, Mont.

"Operations on Finite Automata," *C. C. Elgot and J. D. Rutledge*, IBM Research Center, Yorktown Heights, N. Y.

"Delayed Logic and Finite State Machines," *D. Arden*, Massachusetts Institute of Technology, Cambridge, Mass.

"Techniques for the Diagnosis of Switching Circuit Failures," *J. M. Galey, R. E. Norby, and J. P. Roth*, IBM Corporation, Poughkeepsie, N. Y.

INFORMAL EVENING SESSION—OPEN PROBLEMS IN SWITCHING THEORY

Chairman, C. C. Elgot

Members of the audience will be invited to raise questions concerning open problems in switching theory that they have encountered.

Friday, October 20

LOGIC NETWORK SYNTHESIS

Chairman, J. P. Runyon
Bell Telephone Laboratories
Murray Hill, N. J.

"Canonical Forms of Functions in *P*-Valued Logics," *M. Cohn*, Sperry Rand Research Center, Sudbury, Mass.

"Boolean 2-Terminal Synthesis Based on Incidence Matrices," *S. Okada and K. P. Rajappan*, Stromberg Carlson Co., Research Division, Rochester, N. Y.

"A Computer Program for the Synthesis of Combinational Switching Circuits," *R. M. Karp, F. E. McFarlin, J. P. Roth, and J. R. Wilts*, IBM General Products Division, Endicott, N. Y.

"Automatic Fault Detection in Combinational Switching Networks," *W. H. Kautz*, Stanford Research Institute, Menlo Park, Calif.

PROPOSED GRAPHICAL SYMBOLS FOR DIGITAL COMPUTERS

Chairman, R. G. Preiss
IBM Corporation
Poughkeepsie, N. Y.

"History of Logic Diagramming Standardization and Status Report on ASA Y32.14 'Graphical Symbols for Logic Diagrams'," *T. H. Mott, Jr.*

"Proposed American Standard for Graphical Symbols for Logic Diagrams Y32.14," *A. L. Raiche*, Remington Rand UNIVAC, St. Paul, Minn.

"Usage and Requirements of Standardization for the Military," *W. J. Smith*, Air Force Electronic Systems Division

"Report of Symbology Task Force of AIEE Computer Standards Subcommittee," *R. C. Boden*, IBM Corporation, Kingston, N. Y.

"Survey of Symbols Usage and Preference—A Report by the AIEE Computing Devices Committee," *R. M. Kalb*, Remington Rand UNIVAC, St. Paul, Minn.

"Recognition Characteristics of Uniform and Non-Uniform Shaped Symbols," (speakers to be announced).

1961 EJCC

The 1961 Eastern Joint Computer Conference will be held December 12–14 at the Sheraton-Park Hotel, Washington, D.C. The theme of the conference will be "Computers—Key to Total Systems Control." Papers have been solicited on all advances in computer hardware and concepts leading toward present and future control of industrial, government, defense, and business management systems. The papers to be presented are to be published prior to the conference.

The Preliminary technical program, including abstracts of papers, is as follows:

Tuesday, December 12

TOTAL SYSTEMS IN REAL TIME

Session Chairman: J. H. BURROWS
Mitre Corporation, Bedford, Mass.

Multilevel Programming in a Real-Time System, *A. B. Shafritz amd A. E. Miller*, Auerbach Electronics Corporation, Philadelphia, Pa.

Many complex, modern, real-time systems include, as an integral part, a sophisticated electronic digital data processor. There are usually several, quite diverse, timing requirements imposed on the operational program to meet the real-time deadlines of the system. To obtain efficient processing the program is divided into levels, each level satisfying a different type of system deadline. This paper uses the vehicle of a store-and-forward communication system to demonstrate the power of multilevel programming. The establishment of these levels, their relationship, and criteria for assigning functions to a given level are discussed. The unusual approach for mechanizing the control of transfers between levels is highlighted.

The Development of Computer Programs for the Naval Tactical Data System, *G. G. Chapin and R. A. Hileman*, Remington Rand UNIVAC, San Diego, Calif.

The Naval Tactical Data System (NTDS) employs equipment building blocks. This design approach reduces system development time, simplifies the implementation of different site configurations, and permits many system requirements to be changed without equipment redesign. The real-time data processing problems are shared by a group of physically small, but highly capable, general-purpose computers, Univac AN/USQ-20 Unit Computers. The design and analysis of the NTDS problem is described along with the methods employed by Univac for preparation of operational, test, simulation and support programs. Over 250,000 instructions have been produced for the first two steps of system development.

Multi-Computer Programming for the Naval Tactical Data System, *G. A. Erickson, E. G. Mutschler, and G. E. Pickering*, Remington Rand Univac, San Diego, Calif.

Several multicomputer programs have been designed and implemented by Univac for the real-time Naval Tactical Data System. This paper describes the techniques developed to solve two of the most complex programming problems: Program Control and Inter-Computer Transfer.

System tasks are distributed between Computers. Program Control within each computer considers each assigned task (subprogram) for execution, based on its priority as a system component. Computers are cabled directly together and data and control information are exchanged betwen computer memories.

Inter-computer transfer techniques are applicable to any number of computers, provided the system tasks can be properly divided.

Project Mercury Real-Time Computational and Data-Flow Systems, *S. I. Gass, W. K. Green, J. E. Hamlin, R. Hoffman, R. D. Peavey, A. Peckar, and M. B. Scott*, IBM Federal Systems Division, Washington, D. C.

This paper will be presented in four sections:

The use of digital computers as an integral part of real-time decision making in Project Mercury. Duplexed IBM 7090 computers are used throughout a mission to process observations made in the launch, orbit and, re-entry phases and to supply a continuous, real-time record of the known environmental states of the spacecraft. In addition, a computer-oriented procedure—CADFISS—is used to determine the operational status of world-wide data flow systems prior to a mission.

The Mercury programming system. This system has been built on the modular concept whereby groups of processors which perform various tasks are combined by a monitor program to form a system to handle the demands of any particular phase of the Mercury mission. The techniques employed have required a negligible trade-off between speed and accuracy and have provided the flexibility needed in handling an enormous task with ever-changing specifications.

Simulation procedures. The first method allows the operational programs to run in a quasi-real-time environment under control of a simulation program. The second simulation method allows for the generation of a known set of data which is then processed in real-time by the operational programming system.

Data flow and equipment configurations of the Mercury Launch Monitor Subsystem. The configuration of equipments comprising this system is unique from the standpoint of processing high- and low-speed data transmissions of extreme distances involving computer to computer, computer to special displays and real-time data sources to computer.

SACCS—A Data Processing System for the Strategic Air Command, *E. Wolf and W. S. Ford*, International Electric Corporation, Paramus, N. J.

The SAC requirements for a control system demand an integrated design capable of providing the information and means for CINCSAC to control and command the force with increased effectiveness. This will be accomplished through SACCS, the Strategic Air Command Control System, which automates the routing, processsing, and display of operational data. These functions are performed utilizing the most reliable equipments permitted by the state of the art.

SACCS consists of three subsystems: Data Transmission, Data Processing, and Data Presentation. These subsystems are integrated to perform the SAC planning and control mission through the largest computer program ever undertaken.

Wednesday, December 13

SYSTEMS SIMULATION

Session Chairman: J. SHERMAN
Lockheed Aircraft Corporation
Sunnyvale, Calif.

A Simulation Model for Data System Analysis, *L. Gainen*, The RAND Corporation, Dayton, Ohio.

Designing a data system to support management objectives is, at present, no more than a highly specialized art. Data system designers can bring their profession closer to a predictive science by adapting for data system analysis purposes analytical tools which make possible prediction and quantification of data system behavior.

This paper discusses a generalized data system model and describes a technique of simulating dynamic system operation. Benefits possible through such simulation are explored. The major purpose presently proposed for this analytical technique is to test the feasibility of a data system design before acquisition of actual hardware.

A General Purpose Systems Simula-

tion, *G. Gordon*, IBM Advanced Systems Development Division, Yorktown Heights, N. Y.

The paper discusses a general-purpose program that is being used to carry out simulation studies of systems on a digital computer. The principles upon which the program is based are discussed and a description of the program operation is given. To use the program, the system to be simulated is represented as a block diagram constructed from a set of simple block types. Given such a description, the program automatically prepares a model that simulates the system. The program has proved to be easy to use and has been applied to a wide variety of systems. Experience gained with the program is used to assess the value of general purpose programs in system study work.

Use of a Combined Analog-Digital System for Re-Entry Vehicle Flight Simulation, *Dr. A. Wilson*, General Dynamics—Astronautics, San Diego, Calif.

Simulation of space flight for a cislunar vehicle with re-entry capability is being done at General Dynamics Astronautics on its combined analog-digital system. An unusual feature is the insertion of a pilot and rudimentary cockpit display in the loop. Vehicle dynamics are simulated in real time on a large, general-purpose analog computer. Onboard digital guidance computer is simulated by a digital program on a high-speed digital computer. Closed-loop operation is attained by interconnection through an analog-digital conversion system.

Problems of computer synchronization and control, operating procedures, and results of the simulation will be presented. The problem of error analysis, and the study of simple but definitive test cases for closed-loop simulation will also be discussed.

Combined Analog-Digital Simulation, *A. J. Burns and R. E. Kopp*, Grumman Aircraft Engineering Corporation, Bethpage, N. Y.

Combined analog-digital systems are rapidly becoming operational to overcome the respective limitations of analog and digital computers. One such system which has been used successfully to link an IBM 704 with an analog computer is described.

A missile intercept problem has been solved using analog-digital simulation. Dynamic equations are solved on the analog computer, while guidance and kinematic equations are solved simultaneously on an IBM 704.

Advantages of this type of simulation are illustrated by comparing the accuracy and computing time of the combined analog-digital solution with those of an all-digital and all-analog solution.

CONTRANS (Conceptual Thought Random-Net Simulation), *D. Malin*, Walter Johnson High School, Rockville, Md.

CONTRANS is a computer simulation of a physiologically-oriented reasoning and problem solving model. It utilizes semi-random nets to recognize sensory patterns and attempts to create and manipulate meaningful assemblies of data and to modify its own potential for handling future data. Besides several layers of semi-random nets, the model employes a screen-like memory and representation medium, and several motor functions controlling scanning of the memory.

Input and output consist of modified English sentences. Two kinds of logical operations are employed. One explores the data rapidly for tentative strategies and solutions; the other proceeds by demonstrable intermediate steps, creating progressive intermediate goals.

The Current Status of Programming Language Standardization (A Panel Discussion

Moderator:

H. S. Bright, Data Processing Group, Bema, New York, N. Y.—Chairman ASA X3 Committee (Computers and Data Processing)

Panel Members:

C. A. Phillips, Department of Defense, Washington, D. C.—Chairman, CODASYL Executive Committee

A. J. Perlis, Carnegie Institute of Technology, Pittsburgh, Pa.—Chairman, ACM Language Committee

R. E. Utman, Remington Rand Univac, New York, N. Y.—Chairman *Pro Tempore*, ASA Subcommittee X3.4 (Languages)

R. F. Clippinger, Minneapolis-Honeywell, EDP Division, Wellesley Hills, Mass.—Chairman, ISO/TC 97 Working Group (Languages).

English-based *CO*mmon *B*usiness *O*riented *L*anguage has already been implemented, or soon will be, in processors for most of the U.S.-built commercially-available computers widely used for business data processing, and for several machines of foreign manufacture; *ALGO*rithmic *L*anguage, or dialects of it, for many engineering-scientific machines here and abroad.

ASA Subcommittee X3.4 is working in the U. S., ISO/TC 97, internationally, toward development of generally-acceptable standards in the area of programming languages.

The speakers hold responsible offices in the several organizations concerned and will provide an authoritative snapshot of computer programming language standardization as of December, 1961.

Advances in Equipment

Session Chairman: S. N. Alexander
National Bureau of Standards
Washington, D. C.

Digital to Voice Conversion, *E. Ragland*, Motorola Inc., Military Electronics Division, Chicago, Ill.

This paper presents a report of a new technique for conversion of digital words into vocal words and describes the laboratory experiments and studies directed toward establishment of feasibility of the technique. The conversion method uses photographic memory for the storage of a vocabulary of one to several thousand words. The photographic word is projected onto an electrically charged photosensitive plate for electrostatic readout. The vocabulary is optically scanned at a rate sufficiently high to provide access to any given word in a time imperceptible to the listener. The feasibility experiments performed to establish the operational capabilities of the technique are described in detail. In addition, the program for further development and a report on possible applications are presented.

CRAM (Card Random Access Memory): Functions and Use, *L. Bloom and I. Pardo*, National Cash Register Company, Electronics Division, Hawthorne, Calif.; *W. Keating and E. Mayne*, National Cash Register Company, Data Processing Systems and Sales, Dayton, Ohio.

The National Cash Register Company has developed a new memory configuration which exhibits characteristics heretofore seen only independently in tape systems or random access systems. This paper discusses the specifications of the NCR 315 CRAM system, its application and the programming packages designed for its use.

CRAM'S time-efficiency performance factors as well as its systems versatility are currently unique in the general data processing area. Its uses in file maintenance (random, serial selective and "father-son") as well as sorting are outlined.

The packaged CRAM Executive (PACE) has been programmed to control all CRAM input and output functions, as well as to provide over-all supervision of the entire system. The program retains all the features of earlier NCR Tape Executive Programs.

The Logic Design of the FC-4100 Data Processing System, *W. A. Helbig, A. Schwartz, C. S. Warren, W. E. Woods, and H. S. Zieper*, Radio Corporation of America, West Coast Missile & Surface Radar Division, Van Nuys, Calif.

A general-purpose multiprogram computer has been designed by RCA to meet the need for a compact, reliable, and highly flexible system. This system is capable of handling up to 16 different programs on a sophisticated priority interrupt basis. Each instruction, in addition to performing a conventional operation, may be used to test results for control of short program loops without requiring additional program steps or execution time.

The use of logic on a time-shared basis leads to a considerable saving in hardware, thus achieving high computing ability per dollar invested.

Versatile Man-Machine Communication Console, *R. Green, P. Lazovick, J. Trost, and A. W. Reickord*, RCA Astro-Electronic Products Division, Princeton, N. J.

This paper describes a unique man-computer communication and buffering device designed by the Astro-Electronics Division of RCA. The console allows individuals not trained in machine language to use a computer directly. Translation and machine-language editing are controlled by the console. The time differential between a man's actions and a computer's responses are automatically buffered. The console can be used in a wide variety of information-retrieval and data processing applications.

DATAVIEW, A General Purpose Data Display System, *R. L. Kuehn*, Aeronutronic, Division of Ford Motor Company, Newport Beach, Calif.

In computer-centered complexes, one of the most pressing requirements in recent years has been the presentation of processed data in such a manner as to expedite human comprehension, assimilation and reaction. The latter is particularly true in command and control as well as other decision-oriented

systems. The DATAVIEW system has been developed as a result of extremely stringent operational requirements of a mobile Army headquarters, the Army Tactical Operations Central (ARTOC). Digital signals originated by a central computer are translated with extreme precision and high resolution to multicolored displays of either alphanumeric or graphical content. The methodology of accomplishing this transition to both console size and large screen displays is discussed, including the unique application of the photographic and optical sciences, mechanisms and electronic concepts.

A Computer for Direct Execution of Algorithmic Languages, *J. P. Anderson,* Burroughs Corporation, Paoli, Pa.

A functional design is presented for a computer that directly executes programs written in a particular algorithmic language (ALGOL '60), with no prior translation.

Hardware recognition and interpretation of the various syntactic elements occurring in the string of symbols making up a program, is through the use of push-down lists for the various classes of elements; control, operators, and variables. An address table is employed that relates a variable name (identifier) to a value in a data memory and permits the arbitrary re-use of variable names as specified by the definition of the language. Considerations for other algorithmic languages are given as well as conclusions regarding the future trend of Computer design.

Eddycard Memory—A Semi-Permanent Storage, *T. Ishidate, S. Yoshizawa, and K. Nagamori,* Nippon Electric Company Ltd., Kawasaki, Japan.

The Eddycard memory is a semi-permanent memory device similar to the Unifluxor in principle. However, phenomena are observed from the other point of view, analyzing eddy current distributions in a conductor.

Besides the theoretical analysis, an experimental Eddycard memory with capacity of 1024 words of 45 bits each is introduced in detail. In the experimental memory, information "0" is represented by a square copper film with a punched slot, and "1" by a film without a slot.

The memory works at a cycle time of 100 mμsec, but a 50-mμsec memory will be realized.

Thursday, December 14

COMMUNICATION SYSTEMS

Session Chairman: J. STRONG
North American Aviation
Los Angeles, Calif.

Digital Data Transmission: The User's View, *J. A. Perlman,* Hughes Aircraft Company, Culver City, Calif.

In July, 1960, members of eight major firms in the aerospace industry met to establish an informal Data Transmission Study Group. Objectives of the group were established as exchange of knowledge, standardization of terminology, discussion of total systems requirements, development of reasonably uniform requirements for equipment and service, forecasting of long-range needs, and acting as a coordinating group for these users with the common carriers and equipment manufacturers.

This paper describes the group's results to date in developing composite requirements for intraplant and interplant data collection and transmission systems. Both wireline and microwave systems are included. Projects scheduled for completion in the near future are also discussed.

Tele-Processing Systems, *J. D. Shaver,* IBM Data Systems Division, White Plains, N. Y.

The impact of data transmissions on information processing and the ever-increasing demand of government and industry to utilize transmission facilities in extending the use of computing systems is being reflected in today's data processing equipment development.

The background of development in data transmission devices is discussed as well as accomplishments in this new area. New developments in data handling are explored, indicating the problems encountered in providing a balanced data transmission and processing system. Emphasis is in the data communications area, stressing the business machines supplier's attempts to provide equipment which is compatible with existing and future transmission services and which will also satisfy the operational requirements of the user.

Communications for Computer Applications, *A. A. Alexander,* American Telephone and Telegraph Company, New York, N. Y.

In this paper, some of the interesting applications of communication to computers in associated data processing systems are reviewed. Actual examples of operations and proposed systems are presented.

The major considerations that are involved in determining the type and extent of communications that may be used such as cost, speed, time value, availability and reliability are outlined and discussed.

Some of the newer communications facilities are described, and some thoughts are presented as to the possible future systems applications of communication to computer.

The Saturn Checkout Aystem, *J. Heskin,* Packard Bell Computer Corporation, Los Angeles, Calif.

The checkout procedure for the booster stage of the Saturn vehicle is being automated through the use of a computer-controlled, real-time, closed-loop system.

An expandable multicomputer net allows for the simultaneous running of several test programs for a number of test stations when the vehicle is checked out from the component level to the final over-all system. More than one vehicle can be accommodated at different stages of checkout.

A compiler is provided with a source language, SOL (System Oriented Language), which simplifies the generation of real-time process control programs.

Information Handling in the Defense Communications Control Complex, *T. J. Heckleman and R. H. Lazinski,* Communications Systems Division, Philco Corporation, Fort Washington, Pa.

The Defense National Communications Control Center (DNCCC), recently installed in Washington, is the nerve center of the Defense Communications Control Complex. It provides a real-time on-line system for control and supervision of the worldwide Defense Communications System (DCS), made up of the major military long-haul communications facilities.

Messages concerning the status of communications facilities and traffic backlog in the DCS are inserted directly into a computer at DNCCC, where they are processed and presented on large wall-type displays.

The flow of information through the system is reviewed in detail, with special emphasis on digital data interchange between communications, computer, and displays.

An Automatic Digital Data Acquisition System for Space Surveillance, *M. S. Maxwell,* U. S. Naval Weapons Laboratory, Dahlgren, Va.

An automatic space surveillance system is being developed to detect and determine the orbital elements of unannounced noncooperative satellites as an extension of the present SPASUR system. To do this, digital data from many remote stations are sent continuously at 2500 bits per second to a central assembly system. Selected data are assembled into a form for entry via a tape channel into an IBM 7090 computer with a maximum delay of one second for on-line processing of the data.

Four Advanced Computers—Key to Air Force Digital Data Communications System, *R. J. Segal,* RCA Electronic Data Processing Division, Camden, N. J.

Four computers of advanced design play a central role in the world's largest and most advanced Digital Data Communications System, ComLogNet. The system will initially link 240 Air Commands and other installations.

Two computers (Communication Data Processors), incorporating 1.5-μsec memories, coordinate message flow and sequencing. Two additional computers (Accumulation and Distribution Units) sample incoming channels, perform code conversion, temporarily store information, provide automatic accuracy control, and control outstation and trunk transmissions.

The paper describes how the data communications function is automated. Photographs and a description of the hardware, stressing the unique role of the computers, are presented.

PROGRAMMING AND APPLICATIONS

Session Chairman: C. A. PHILLIPS
Department of Defense
Washington, D. C.

The Atlas Supervisor, *T. Kilburn and R. B. Payne,* The University of Manchester, Manchester, England; *D. J. Howarth,* Ferranti Ltd., London, England.

All activities of the Atlas computer are controlled by the supervisor programs within the computer. Basic protection is provided by hardware, but extra complete protection is provided by the routines themselves. Over-all control of jobs is designed to maintain the highest possible activity of all parts of the computing system. The allocation and addressing of storage is done by the supervisor, through which the operation of tapes and other peripheral equipment is organized.

A Syntax Directed Generator, *S. War-*

shall, Computer Associates, Inc., Woborn, Mass.

A novel technique of compiler organization permits tabular description of not only the source language, but also the target machine. A short, universal algorithm—independent of both language and machine—uses instances of such tables to effect translation. Since a compiler for a new language and a new machine is simply a new set of tables, compiler construction cost may frequently be cut to a matter of man-weeks of effort. The unusual organization of the compiler makes it a useful tool in the translation of languages which are not completely formal, and also permits very inexpensive embedding of the compiler in an over-all programming system.

An Automated Technique for Conducting a Total System Study, *A. O. Ridgway,* IBM Federal Systems Division, Bethesda, Md.

In conjunction with a comprehensive analysis of the data processing at an Air Force Base, a system study technique was developed which utilized EAM equipment for recording and analyzing all primary data processing applications. The study effort also included the initial postulation of a total data processing system.

The automated study technique facilitated systems analysis by: automatically relating significant data, providing thorough analyses of existing systems, simplifying documentation correction, and automatically preparing flow charts.

The study technique is completely independent of the data processing complex being analyzed, the organization involved, the mission it supports, and hardware considerations.

Display System Design Considerations, *R. T. Loewe and P. Horowitz,* Aeronutronic, Division of Ford Motor Company, Newport Beach, Calif.

Several significant system concepts and human engineering factors related to display systems are discussed. These include: amount of information displayed, group vs individual displays, response times, retinal resolution, display detail and size, audience configuration and viewing environment. Types of formats used for presenting information are summarized. Display system evaluation criteria and a checklist of primary display system requirements and characteristics are discussed.

Abstract Shape Recognition by Machine, *M. E. Stevens,* National Bureau of Standards, Washington, D. C.

Graphic pattern recognition in development of total systems for information selection and retrieval is considered. A machine model for abstract shape recognition is described. Examples are given of recognition of 15 to 20 categories of geometric shapes. Using a contour-projection principle, the model identifies various graphic patterns, regardless of size and locational transformations. It provides means for predicting whether large classes of patterns would be confused with any of the patterns that are recognizable. Possibilities for recognition of of constrained handdrawn figures, such as chemical structure diagrams, are considered. Possible means of instrumentation are also discussed.

A Character Recognition System Which is Independent of Character Translation and Orientation, *D. N. Buell,* Chrysler Corporation, Centerline, Mich.

A character recognition system is described which incorporates a lens-and-retina input and a relatively simple computer. The system operates effectively with the image focused anywhere on the retina and may operate so as to be independent of image orientation.

The characters which may be unambiguously discriminated are those which have distinct transform functions T^*. A simple algorithm is given for obtaining T^* for both curvilinear and rectilinear figures. Examples are given and possible means of resolving the ambiguities are discussed.

The Digital Computer as an Aid in the Diagnosis of Heart Disease, *C. A. Steinberg and W. E. Tolles,* Airborne Instruments Laboratory, Deer Park, L. I., N. Y.; *A. H. Freeman, M.D.,* Sloan-Kettering Memorial Cancer Center, New York, N. Y.; *C. A. Caceres, M.D., and S. Abraham,* U. S. Public Health Service, Washington, D. C.

The rapid computational capabilities and large storage capacity of the digital computer provides the physician with a powerful tool for analyzing physiological waveforms of the heart and their relationship to cardiovascular pathology. A pattern recognition program for automatically recognizing clinically useful parameters from physiological waveforms has been successfully developed. These parameters from the physiological waveforms were then combined into a multidimensional probability distribution. Different distributions were formulated and stored in the computer for the normal and pathological groups of patients. The compatibility of a patient's parameters with those stored in the computer was calculated and can be used as a diagnostic aid.

COMING MEETINGS—PAPERS DEADLINE AHEAD

AIEE WINTER MEETING

At the AIEE Winter General Meeting in New York City, January 28–February 2, 1962, the AIEE Computer Systems Subcommittee will sponsor sessions on Kilomegacycle (Gigacycle) Computing Systems (defined simply as systems operating at clock frequencies near and above 1000 Mc). Authors who have already submitted abstracts and summaries are urged to observe the following deadlines.

October 30, 1961

Deadline for full text of TRANSACTIONS papers prepared in accordance with the AIEE authors' guide, one copy to be sent to the papers chairman, and four to Mr. E. C. Day, Assistant Secretary for Technical Papers, American Institute of Electrical Engineers, 33 West 39th Street, New York 18, N. Y.

November 25, 1961

Up to this date, papers can be accepted as "Conference Papers" for preprint, but cannot be immediately processed for TRANSACTIONS. Copies are to be sent to Mr. Day and the papers chairman as above.

Correspondence regarding these sessions should be addressed to:

J. H. Wright, *Papers Chairman*
Division 12
National Bureau of Standards
Washington 25, D. C.

IFIPS CONGRESS '62

The International Federation of Information Processing Societies (IFIPS) will hold a Congress in Munich, Germany, from August 27 to September 1, 1962. The Congress will cover all aspects of Information Processing and Digital Computers including the following:

1) Business Information Processing: Data processing in commerce, industry, and administration.

2) Scientific Information Processing: Numerical analysis; calculations in applied mathematics, statistics, and engineering; data reduction; problems in operations research.

3) Real Time Information Processing: Reservation systems; computer control; traffic control; analog-digital conversion.

4) Storage and Retrieval of Information: Memory devices; library catalogs.

5) Language Translation and Linguistic Analysis.

6) Digital Communication: Encoding; decoding; error-detecting and error-correcting codes for digital data transmission.

7) Artificial Perception and Intelligence: Pattern recognition; biological models; machine learning, automata theory.

8) Advanced Computer Techniques: Logical design; logical elements, storage devices; ultra-high-speed computers; program techniques; ALGOL.

9) Education: Selection and training of computer specialists; training of nonspecialists in the use of computers; information processing as a university subject.

10) Miscellaneous Subjects: Growth of the information-processing field.

In each category, it is planned to cover, where appropriate, the applications of digital computers, programming, systems design, logical design, equipment, and components.

U. S. authors wishing to offer papers are invited to send abstracts of 500–1000 words to:

Dr. E. L. Harder
Westinghouse Electric Corporation
East Pittsburgh, Pa.,

by September 15, 1961. These abstracts will be considered by the international program committee of IFIPS, and authors of selected abstracts will be invited to submit their complete papers (in French or English) for consideration by the program committee in March, 1962. Authors in other countries should send their manuscripts to the national representative for their country.

In addition to contributed papers, there will be invited papers, symposia, and panel discussions.

1962 SPRING JOINT COMPUTER CONFERENCE CALL FOR PAPERS

This is the first call for papers for the 1962 Spring Joint Computer Conference, to be held in San Francisco, Calif., on May

1–3, 1962. This conference is the direct successor to the Western Joint Computer Conferences of previous years. The new name reflects the desire of the new sponsoring organization, the American Federation of Information Processing Societies (AFIPS), to designate the conferences by their times rather than by their locations.

The Conference Committee has decided to dispense with an official theme or slogan. Nevertheless, the implicit theme remains. The objective is, as always—to publish, distribute, present, and discuss new and significant information on achievements, trends, concepts and techniques in the computer field and allied areas.

Evaluation of the submitted papers will be based on a review of the *complete preliminary draft* of each paper. We request that the paper be complete, to enable full consideration of the technical content. However, as a preliminary draft, the text and drawings need only be clear and readable, not necessarily formal or artistic.

To enable adequate review of your paper, and to permit distribution of the *Conference Proceedings* at the time of registration, please submit three copies of your paper to the Technical Program Committee as soon as possible, or by November 10, 1961, at the latest. The papers will be reviewed, final selections made, and authors notified by January 19, 1962. The final drafts of selected papers must then be received by March 1, 1962, for inclusion in the *Conference Proceedings*.

As in the past, the basic quality and value of the technical sessions will depend upon your submitted papers. The Committee earnestly solicits your cooperation and support.

Richard I. Tanaka, *Chairman*
Technical Program Committee
1962 Spring Joint Computer Conf.
Lockheed Missiles and Space Co.
3251 Hanover Street
Palo Alto, Calif.

PUBLICATIONS AVAILABLE

N.S.F. Report on "Current Research and Development in Scientific Documentation"

The National Science Foundation has released the eighth report in the series entitled "Current Research and Development in Scientific Documentation," a semi-annual publication which contains descriptions of research and development projects in the field of scientific documentation and potentially related areas. Issue No. 8 includes descriptions of 195 research projects in 122 organizations.

Included are all pertinent activities on which information could be obtained in the United States and in 16 foreign countries. For the first time since the report series was begun in 1957, the Foundation received a number of descriptive statements on research being conducted in the Soviet Union. The first Moscow State Pedagogical Institute of Foreign Languages, the Institute of Electronics, Automatics and Telemechanics, and the Leningrad State University submitted descriptions of their research in mechanical translation. The other 15 foreign countries included in the report are: Belgium, Czechoslovakia, France, Germany, Great Britain, Hungary, India, Italy, Japan, Lebanon, Mexico, The Netherlands, Rumania, Sweden, and Yugoslavia.

The report includes descriptions of projects concerned with scientists' information needs, the uses made of scientific information, communication problems in science and technology, new methods for the storage and retrieval of information, mechanical translation of languages, and the design of equipment for processing scientific information. Also included are descriptions of research on problems not immediately connected with scientific documentation but whose solution is likely to have an impact on the future of documentation, including studies in such areas as character and pattern recognition, speech analysis and synthesis, linguistic analysis and lexicography, artificial intelligence, and psychology.

Copies of "Current Research and Development in Scientific Documentation, No. 8" may be obtained from the Superintendent of Documents, U. S. Government Printing Office, Washington 25, D. C. (Price, 65 cents.)

INFORMATION FOR AUTHORS

IRE TRANSACTIONS ON ELECTRONIC COMPUTERS is published quarterly, in March, June, September, and December, with a distribution of over 9000 copies, largely to engineers, logicians, and supervisors in the computer field. Its scope includes the design, theory, and practice of electronic computers and data-processing machines, digital and analog, and parts of certain related disciplines such as switching theory and pulse circuits.

If a paper of widespread interest beyond the computer field is submitted, it will be recommended to the Editor of PROCEEDINGS OF THE IRE for publication. If our reviewers feel that a paper should be submitted to a different IRE TRANSACTIONS, we will so recommend to the author.

Publication time in IRE TRANSACTIONS ON ELECTRONIC COMPUTERS, from receipt of the original manuscript to mailing of the issue, is normally in excess of 5 months, but can be made as little as 3½ months if the occasion demands and the manuscript is carefully prepared.

To avoid delay, please be guided by the following suggestions:

A. Process for Submission of a Technical Paper

1) Send to the appropriate Editor three copies of your manuscript, each copy complete with illustrations. (For Letters to the Editor, two copies will do.)

2) Enclose originals for the illustrations, in the style described below. Alternatively, be ready to send the originals immediately upon acceptance of the paper.

3) Enclose a separate sheet giving your preferred address for correspondence and return of proofs.

4) Enclose a technical biography and photograph of each author, or be ready to supply these upon acceptance of the paper. For biography style, see any IRE journal.

5) If the manuscript has been presented, published, or submitted for publication elsewhere, please so inform the Editor. Our primary objective is to publish technical material not available elsewhere, but on occasion we publish papers of unusual merit that have appeared or will appear before other audiences.

B. Style for Manuscript

1) Typewrite, double or 1½ space; use one side of sheet only. (Good office-duplicated copies are acceptable.)

2) Provide an informative 100- to 250-word summary (abstract) at the head of the manuscript. It will appear with the paper and also separately in PROCEEDINGS OF THE IRE.

3) Provide a separate double-spaced sheet listing all footnotes, beginning with "*Received by the PGEC _____," and "†(Affiliation of author)," and continuing with numbered references. Acknowledgment of financial support is often placed at the end of the asterisk footnote.

4) References may appear as numbered footnotes, or in a separate bibliography at the end of the paper, with items referred to by numerals in square brackets, *e.g.*, [12]. In either case, references should be complete, and in IRE style.

Style for papers: Author (with initials first), title, journal title, volume number, inclusive page numbers; month, year.

Style for books: Author, title, publisher, location, year; page or chapter numbers (if desired).

See this or previous issues for further examples.

5) Provide a separate sheet listing all figure captions, in proper style for the typesetter, *e.g.*: "Fig. 1—Example of a disjoint and distraught manifold."

C. Style for Illustrations

1) Originals for illustrations should be sharp, noise-free, and of good contrast. We regret that we cannot provide drafting or art service.

2) Line drawings should be in India ink on drafting cloth, paper, or board. Use 8½×11 inch size sheets if possible, to simplify handling of the manuscript.

3) On graphs, show only the coordinate axes, or at most the major grid lines, to avoid a dense, hard-to-read result.

4) All lettering should be large enough to permit legible reduction of the figure to column width, perhaps as much as 4:1.

5) Photographs should be glossy prints, of good contrast and gradation, and any reasonable size.

6) Number each original on the back, or at the bottom of the front.

7) Note item B-5 above. Captions lettered on figures will be blocked out in reproduction, in favor of typeset captions.

Mail analog and hybrid computer manuscripts to:
John E. Sherman
Associate Editor, IRETEC
Lockheed MSD
Sunnyvale, Calif.

Mail logic and switching theory manuscripts to:
Prof. E. J. McCluskey
Associate Editor, IRETEC
Dept. of Electrical Engineering
Princeton University
Princeton, N. J.

Mail all other manuscripts to:
Prof. Norman R. Scott
Editor-in-Chief, IRETEC
Dept. of Electrical Engineering
University of Michigan
Ann Arbor, Mich.

(*Continued from outside back cover*)

IRE TRANSACTIONS ON ELECTRONIC COMPUTERS

Volume EC-10 **SEPTEMBER, 1961** **Number 3**

Reviews of Books and Papers in the Computer Field

(See front cover for main contents of issue)

(Continued on inside back cover)